



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA PODNIKATELSKÁ  
ÚSTAV INFORMATIKY

FACULTY OF BUSINESS AND MANAGEMENT  
INSTITUTE OF INFORMATICS

## NÁVRH NABÍDKOVÉHO SYSTÉMU PRO CESTOVNÍ AGENTURU

DESIGN OF A TRAVEL AGENCY OFFERING SYSTEM

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

ZBYNĚK PETR

VEDOUCÍ PRÁCE  
SUPERVISOR

Ing. ALEŠ KLUSÁK, Ph.D.

BRNO 2015

Tato verze bakalářské práce je zkrácená (dle Směrnice děkana č. 2/2013). Neobsahuje identifikaci subjektu, u kterého byla bakalářská práce zpracována (dále jen „dotčený subjekt“) a dále informace, které jsou dle rozhodnutí dotčeného subjektu jeho obchodním tajemstvím či utajovanými informacemi.

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

**Zbyněk Petr**

Manažerská informatika (6209R021)

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách, Studijním a zkušebním řádem VUT v Brně a Směrnicí děkana pro realizaci bakalářských, magisterských a doktorských studijních programů zadává bakalářskou práci s názvem:

### **Návrh nabídkového systému pro cestovní agenturu**

v anglickém jazyce:

### **Design of a Travel Agency Offering System**

Pokyny pro vypracování:

Úvod  
Vymezení problému a cíle práce  
Teoretická východiska práce  
Analýza problému a současné situace  
Vlastní návrhy řešení, přínos návrhů řešení  
Závěr  
Seznam použité literatury  
Přílohy

Podle § 60 zákona č. 121/2000 Sb. (autorský zákon) v platném znění, je tato práce "Školním dílem". Využití této práce se řídí právním režimem autorského zákona. Citace povoluje Fakulta podnikatelská Vysokého učení technického v Brně. Podmínkou externího využití této práce je uzavření "Licenční smlouvy" dle autorského zákona.

Seznam odborné literatury:

- BUDD, A., M. CAMERON and S. COLLISON. CSS Mastery. Advanced Web Standards Solutions. New ed. Berkeley [Calif.]: Apress, 2006. ISBN 15-905-9614-5.
- ERL, T. SOA: servisně orientovaná architektura: kompletní průvodce. Překlad Ondřej BAŠE a Lukáš KREJČÍ. Brno: Computer Press, 2009. ISBN 978-80-251-1886-3.
- GUTMANS, A. Mistrovství v PHP 5. Brno: Computer Press, 2007. ISBN 978-80-251-1519-0.
- KEITH, J. HTML5 for Web Designers. New York: A Book Apart, 2010. ISBN 978-0-9844425-0-8.
- KOFLER, M. Mistrovství v MySQL 5. Překlad Jan SVOBODA, Ondřej BAŠE a Jaroslav ČERNÝ. Brno: Computer Press, 2007. ISBN 978-80-251-1502-2.

Vedoucí bakalářské práce: Ing. Aleš Klusák, Ph.D.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2014/15.



*B. Půža*

doc. RNDr. Bedřich Půža, CSc.  
Ředitel ústavu

*Stanislav Škapa*

doc. Ing. et Ing. Stanislav Škapa, Ph.D.  
Děkan

V Brně, dne 28. 2. 2015

## **Abstrakt**

Cílem této bakalářské práce je navrhnout informační systém pro vytváření nabídek zájezdů klientům cestovní agentury. Systém bude integrován do webového portálu a bude umožňovat interaktivní formou vytvářet pro zákazníky nabídky zájezdů na míru a také s nimi komunikovat.

## **Abstract**

The subject of this bachelor's thesis is to design an information system for creation of the offers for customers of the travel agency. The system will be integrated into web portal and will allow to create offers in interactive way and also to communicate with customers.

## **Klíčová slova**

Informační systém, nabídkový systém, procesy, HTML5, JavaScript, jQuery, webové služby CSS, PHP, MySQL, AJAX

## **Key words**

Information system, offering system, processes, HTML5, JavaScript, jQuery, web services, CSS, PHP, MySQL, AJAX

## **Bibliografická citace mé práce:**

PETR, Z. Návrh nabídkového systému pro cestovní agenturu. Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2015. 67 s. Vedoucí bakalářské práce Ing. Aleš Klusák, Ph.D.

### **Čestné prohlášení**

Prohlašuji, že předložená bakalářská práce je původní a zpracoval jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem ve své práci neporušil autorská práva (ve smyslu Zákona č. 121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským).

V Brně dne 2.6.2015

.....

podpis studenta

## **Poděkování**

Rád bych poděkoval vedoucímu práce Ing. Aleši Klusákovi, Ph.D. za vedení práce.

Velký dík patří mé manželce za její podporu a trpělivost.

## Obsah

ÚVOD.....	9
Základní popis firmy.....	9
CÍL PRÁCE.....	10
1 Teoretická východiska práce.....	11
1.1 OOP (Object Oriented Programming).....	11
1.2 PHP.....	12
1.2.1 PHP a OOP.....	12
1.3 Framework.....	16
1.4 Zend Framework (ZF).....	17
1.4.1 Adresářová struktura projektu v ZF1.....	17
1.4.2 Konvence a názvosloví.....	19
1.4.3 Komponenty Zend Frameworku.....	19
1.5 MySQL.....	21
1.6 Webové služby.....	21
1.6.1 SOAP protokol.....	22
1.6.2 WSDL.....	25
1.7 Návrhové vzory.....	25
1.8 HTML.....	26
1.9 CSS.....	27
1.10 JavaScript.....	28
1.11 jQUERY.....	29
1.12 GIT.....	30
2 Analýza současného stavu.....	31



2.1 Popis firmy.....	31
2.2 Popis procesů.....	34
2.2.1 Prodejní proces.....	34
2.3 SWOT analýza.....	37
3 Návrh řešení.....	40
3.1 Analýza.....	40
3.1.1 Kontextový diagram.....	41
3.1.2 Databáze.....	42
3.1.3 Návrh tříd.....	46
3.2 Implementace.....	49
3.2.1 Vytvoření nabídky.....	49
3.2.2 Editace nabídky.....	50
3.2.3 Odeslání nabídky.....	52
3.2.4 Zobrazení nabídky zákazníkem.....	54
3.2.5 Reakce na nabídku.....	55
4 Ekonomické zhodnocení.....	57
4.1 Náklady.....	57
4.2 Přínosy.....	58
ZÁVĚR.....	60
SEZNAM OBRÁZKŮ.....	65
SEZNAM TABULEK.....	66
SEZNAM POUŽITÝCH ZKRATEK A SYMBOLŮ.....	67

# ÚVOD

## **Základní popis firmy**

Cestovní agentura - online prodejce zájezdů v České republice.

K cílům společnosti patří primárně spokojenost zákazníků s poskytovanými službami. Proto si společnost snaží vybírat ke spolupráci zavedené a přední cestovní kanceláře z českého i německého trhu. Zájezdy nabízí nejen online přes své webové stránky, ale také pomocí affiliate marketingu (partnerský marketing). Jde o systém založený na propagaci produktu firmy prostřednictvím stránek affiliate partnerů, kteří za to dostávají určitý podíl z prodeje neboli provizi.<sup>1</sup> Další způsob prodeje zájezdů jsou kamenné prodejny, nejčastěji formou tzv. franchis. Tyto prodejny jsou vybírány velmi přísně.

Společnost zaměstnává 150 – 300 lidí. Nejvíce zaměstnanců má v období duben - říjen, kdy je hlavní sezóna prodeje zájezdů. Společnost se skládá z několika oddělení: marketingové, personální, finanční, call centrum, programátorské, IT, obchodní.

---

<sup>1</sup> <http://www.adaptic.cz/znalosti/slovnicek/affiliate-marketing/>

## CÍL PRÁCE

Cílem této bakalářské práce je analyzovat a navrhnout online nabídkový systém, který bude sloužit pro efektivní komunikaci prodejců a pracovníků firmy se zákazníkem. Systém má umět sestavit nabídku zájezdů zákazníkovi podle jeho požadavků tak, aby v ní přehledně viděl všechny informace týkající se zájezdu, cenové kalkulace jednotlivých nabízených variant a produktů včetně fotografií a recenzí, pokud jsou k dispozici.

Systém má být atraktivní pro používání, interaktivní a přehledný, aby se s ním prodejci i zákazníkovi dobře pracovalo. Zároveň jeho interaktivita má co nejvíce usnadnit prodejci vytváření těchto nabídek a systém mu má co nejvíce šetřit čas a minimalizovat množství jím zadávaných údajů.

Nabídkový systém by měl být do značné míry samostatný, ale musí úzce spolupracovat s ostatními firemními systémy jako jsou interní prodejní informační systém, web a databáze zájezdů. Z prodejního informačního systému nabídkový systém získá údaje o zákazníkovi a informace o jeho požadavcích. Z databáze zájezdů získá informace o zájezdu, ceníkové položky zájezdu a další informace. Z webového portálu potom bude využívat některé jeho funkce tak, aby do něj byl vizuálně integrován.

Nabídkový systém by měl splňovat zásady správného návrhu IS a pro jeho implementaci musí být použity vhodně zvolené moderní technologie. Přednostně technologie, na kterých standardně běží i další systémy, se kterými nabídkový systém bude komunikovat. Při implementaci je kladen důraz zejména na přehlednost celého řešení, které následně vede ke snadnější správě a modifikaci systému v rámci týmové spolupráce ostatních programátorů.

V teoretické části práce popíšu zvolené technologie pro implementaci nabídkového systému, v další části se zaměřím na současný stav firmy a ve třetí části této práce potom navrhnu konkrétní řešení a provedu analýzu celého systému, který bude implementován ve firmě.

# 1 Teoretická východiska práce

## 1.1 OOP (Object Oriented Programming)

Jedná se o moderní styl programování, který je dnes již standardem pro návrh a vývoj většiny systémů. Pro použití těchto technik musí daný programovací jazyk OOP podporovat. Základním prvkem OOP je tzv. objekt. Objekt je ucelená jednotka, která skrývá svůj vnitřní stav a je možné s ním pracovat pomocí veřejných funkcí či procedur, které v OOP nazýváme metody. Při práci s objektem využíváme veřejného rozhraní, které nám objekt poskytuje, a tak nemusíme znát jeho vnitřní strukturu ani konkrétní implementaci.<sup>2</sup>

Objekty lze vzájemně uvádět do různých vztahů - skládat je dohromady, vytvářet hierarchie apod. Tyto vztahy objektů, které jsou obecně využívány pro různé druhy úloh při programování systémů, popisují tzv. návrhové vzory.<sup>3</sup>

Základní pojmy v OOP<sup>4</sup>:

- **Objekt** (object) - ucelená jednotka obsahující vnitřní stav (reprezentovaný tzv. atributy), vnitřní chování (reprezentované tzv. metodami) a protokol zpráv (udává veřejné rozhraní).
- **Třída** (class) - jedná se o formální schéma objektu, které předepisuje vzor objektu, jak z pohledu vnitřní struktury, tak jeho veřejné rozhraní. Na základě tohoto přepisu je potom možné vytvářet objekty.
- **Instance třídy** (instance) je konkrétní vytvořený objekt na základě třídy, který má své vlastní místo v paměti počítače. Má svůj vlastní vnitřní stav.
- **Atributy** (attributes) jsou vnitřní proměnné objektu, se kterými objekt pracuje pomocí metod, a které popisují jeho vnitřní stav.
- **Metody** (methods) - nazýváme tak vnitřní funkce či procedury objektu, které objekt a jeho vnitřní stav modifikují nebo provádějí určité operace.

Mezi základní principy OOP patří zapouzdřenost, dědičnost a polymorfismus.

---

<sup>2</sup> <http://voho.cz/wiki/oop/>

<sup>3</sup> <http://voho.cz/wiki/oop/>

<sup>4</sup> <http://voho.cz/wiki/oop/>

## 1.2 PHP

*“PHP je skriptovací jazyk a byl vytvořen právě pro web a umístění na straně serveru. Do své stránky můžete umístit PHP kód, který se vykonává pokaždé, když má být stránka zobrazena. Váš PHP kód je přeložen webovým serverem a generuje HTML nebo jiný výstup, který pak uvidí uživatel.”*<sup>5</sup>

Takto v začátku své knihy popsal Luke Welling jazyk PHP v roce 2002, kdy tento jazyk dosáhl verze 4. Dnes bychom si s takto jednoduchou definicí jazyka PHP již nevystačili. Za dalších 13 let vývoje totiž prošel obrovskými změnami a dnes již můžeme o PHP mluvit jako o plnohodnotném programovacím jazyce, který se ve většině aspektů vyrovná svým konkurentům.

PHP původně vytvořil Rasmus Lerdorf v roce 1994. V roce 2001 byl jazyk nasazen již na skoro 5 mil. domén, v roce 2010 překročilo nasazení 100 mil. domén po celém světě.<sup>6</sup> Ve světě webových stránek a aplikací se tak jedná o jeden z nejrozšířenějších jazyků. PHP je open source produkt. To znamená, že máme přístup ke zdrojovému kódu. Můžeme ho používat, upravovat a dále distribuovat – a to vše bez jakýchkoliv poplatků.<sup>7</sup>

PHP je tzv. skriptovací jazyk. Jeho kód tedy není kompilován do binární podoby jednorázově při sestavení programu, ale tento zdrojový kód je prováděn interpretem jazyka a zpracováván při každém požadavku na spuštění. Patří tedy do stejné rodiny interpretovaných serverových programovacích jazyků jako je např. Python, Ruby nebo Bash.

### 1.2.1 PHP a OOP

V roce 2004 byla uvolněna přelomová verze 5.0. Řada 5 přinesla zejména nové objektově orientované funkce do jazyka. Mezi novinky patřilo zejména:<sup>8</sup>

---

<sup>5</sup> WELLING, Luke. PHP a MySQL - rozvoj webových aplikací. Praha, 2002, s. 28

<sup>6</sup> <http://php.net/usage.php>

<sup>7</sup> WELLING, Luke. PHP a MySQL - rozvoj webových aplikací. Praha, s. 28

<sup>8</sup> GUTMANS, Andi. Mistrovství v PHP 5, Brno, 2007, s. 35-43

1. Modifikátory *public*, *private*, *protected*
  - Definují u atributů a metod objektu jejich viditelnost (vně objektu, pouze uvnitř objektu, pouze pro potomky dané třídy).
2. Unifikovaný název konstruktoru `__construct()`
  - Konstruktore je speciální vnitřní metoda objektu, která je automaticky provedena při vytvoření objektu.
3. Destrukce objektů `__destruct()`
  - Destruktor je speciální vnitřní metoda objektu, která je automaticky provedena při zahození objektu.
4. Rozhraní
  - Předepisuje veřejné metody, které objekt daného typu musí implementovat. Jedno rozhraní může být implementováno více třídami. Víme potom, že pokud máme třídu daného typu (tzn. implementující dané rozhraní), můžeme při práci s ní využít metody předepsané v tomto rozhraní a nezajímá nás konkrétní implementace.
5. Operátor *instanceof*
  - Slouží pro testování, zda je objekt instancí určité třídy.
6. Konečné metody (*final*)
  - Není možné přetěžovat v děděných třídách.
  - Klíčové slovo *final*.
7. Konečné třídy
  - Není možné z nich dále děděním vytvářet potomky.
  - Klíčové slovo *final*.
8. Klonování objektů
  - Vytvoří novou instanci třídy se stejným aktuálním vnitřním stavem.
  - Klíčové slovo *clone*.
9. Konstanty třídy
  - Neměnné pojmenované hodnoty, které mohou být využívány uvnitř třídy i mimo ni.
  - Klíčové slovo *const*.

#### 10. Statické metody

- Neváží se na konkrétní instanci třídy. Lze je volat bez vytvoření objektu.
- Klíčové slovo *static*.

#### 11. Statické členy

- Neváží se na konkrétní instanci třídy, ale jsou společné pro všechny vytvořené instance třídy.
- Klíčové slovo *static*.

#### 12. Abstraktní třídy

- Z těchto tříd samotných nelze vytvořit instanci. Objekt je možné vytvořit až z poděděné třídy.
- Klíčové slovo *abstract*.

#### 13. Abstraktní metody

- Abstraktní třída může obsahovat i abstraktní metody. Tzn. tyto metody nejsou implementovány v samotné abstraktní třídě, ale až v jejích potomcích.

#### 14. Informace o typech

- Deklarace funkcí a metod mohou obsahovat informace o datových typech jednotlivých parametrů.

#### 15. Method chaining

- Pokud metoda třídy vrací celý samotný objekt, je možné na něm opět zavolat jeho další metodu.

#### 16. Iterátory

- Implementace rozhraní, pomocí kterého je možné procházet položky objektu v cyklu *foreach*.

#### 17. Metoda `__autoload()`

- Slouží pro automatické zahrnutí zdrojových souborů potřebných pro běh programu. Pokud se snažíme vytvořit instanci třídy, která dosud nebyla deklarovaná, potom se zavolá funkce `__autoload()`, která potřebné soubory do programu může zahrnout.

#### 18. Ošetřování výjimek

- Část zdrojového kódu může být uzavřena do tzv. *try* bloku, ze kterého je možné vyvolat výjimku v případě neočekávaného stavu v programu.

Výjimka musí být instancí třídy *Exception* nebo jejích potomků. Tuto výjimku následně odchytíme v *catch* bloku, kde na ni můžeme patřičně zareagovat.

#### 19. *Foreach* s referencí

- Při procházení pole můžeme jeho jednotlivé položky procházet tzv. s referencí. To znamená, že pokud proměnnou dané položky v cyklu modifikujeme, bude změna provedena přímo v původním procházeném poli.

#### 20. Implicitní hodnoty parametrů předávaných odkazem

- Některé parametry funkcí (uvedené na konci v deklaraci funkce) mohou být pro funkci volitelné. Tzn., že pokud je funkci nepředáme, použije funkce vhodné implicitní hodnoty těchto parametrů. Po volitelném parametru již nesmí následovat povinný parametr.

#### 21. Inovace možností zpracování XML

- Zejména rozhraní SimpleXML, které umožňuje pracovat s XML dokumentem jako s objektem v PHP.

#### 22. SOAP

- Díky dobré podpoře zpracování XML také do PHP přichází podpora protokolu SOAP, který je na XML založen. Jazyk se tak stává vhodným nástrojem pro implementaci webových služeb.

Mezi současné, plně podporované verze PHP patří verze 5.4 a 5.5.

Příklad zdrojového kódu s využitím OOP v PHP 5:

```
interface Storage {
```

```
    public function loadData();
```

```
    public function saveData($data);
```

```
}
```

```
abstract class StringDataStorage implements Storage {
```

```
    protected $data = null;
```

```
    public function getDataSize() {
```

```
        if ($data === null) {
```

```
            throw new Exception("Data hasn't been loaded!");
```

```
        }
```



```

        return strlen($data);
    }
    abstract function loadData();
    abstract function saveData($data);
}
class StringFileStorage extends StringDataStorage {
    private $filename = "";
    public function __construct($filename) {
        $this->filename = $filename;
    }
    public function loadData() {
        $this->data = file_get_contents($this->filename);
        return $this->data;
    }
    public function saveData($data) {
        return file_put_contents($this->filename, $data);
    }
}

```

### 1.3 Framework

Anglické slovo framework má v češtině významy jako kostra, rámec, struktura nebo skelet. V podstatě tak vystihuje i jeho význam a úlohu při vývoji software (SW). Framework je připravená část zdrojových kódů, knihoven a jiných obecných částí aplikace, která je uspořádána do logické struktury a navržena pro obecné využití v dané oblasti. Tuto strukturu potom programy postavené nad tímto frameworkem dodržují. Cílem frameworků je specializovat se na konkrétní oblast vývoje software či technologie (např. webové aplikace, mobilní aplikace) a poskytnout kvalitní základ pro stavbu aplikace tohoto druhu, aniž by programátor musel složitě řešit elementární, často se vyskytující programátorské problémy opakovaně. Tím se výsledné řešení stává přehlednější a čitelnější. Navíc frameworky často používají různé návrhové vzory a “best practices” pro vývoj v dané oblasti, a to zejména díky tomu, že za jejich tvorbou stojí většinou velice zkušení programátoři či celé organizace.

## 1.4 Zend Framework (ZF)

Zend Framework je určen pro vývoj webových aplikací s použitím jazyka PHP 5. Je napsán čistě s použitím objektově orientovaného programování. Přestože jednotlivé komponenty ZF spolu mohou spolupracovat, čímž se stává framework velice výkonný, každá komponenta může být použita také samostatně, nezávisle na ostatních. Aplikace založené na Zend Frameworku jsou snadno udržitelné a rozšiřitelné. Zend Framework nabízí vysoký výkon a při implementaci svých tříd používá hned několik návrhových vzorů. Poskytuje výborný základ pro webové aplikace založené na architektuře model-view-controller (MVC). Hlavní společnost stojící za vývojem Zend Frameworku se jmenuje Zend Technologies, podporují ho však i jiné společnosti jako například Google či Microsoft.<sup>9</sup>

Aktuálně je udržován Zend Framework ve dvou verzích - původní ZF1, jehož poslední release nese označení 1.12 a nedávno vydaný ZF2, jehož aktuální vydání nese označení 2.3. ZF1 a ZF2 nejsou vzájemně kompatibilní, používají jinou strukturu projektu.

ZF2 využívá nové vlastnosti objektově orientovaného programování, které s sebou přineslo až PHP 5.3. Přestože Zend Framework 2 si stáhlo již 15 miliónů uživatelů, Zend Framework 1 zůstává nadále významným produktem, jelikož za dobu jeho existence od roku 2007 na jeho základu běží nespočet webových aplikací.<sup>10</sup>

Mezi výhody Zend Frameworku obou verzí patří fakt, že za nimi stojí významná společnost, která se zabývá vývojem produktů pro vývoj v PHP. Jedná se o celosvětově využívaný framework s velkou komunitou vývojářů a výbornou dokumentací.

### 1.4.1 Adresářová struktura projektu v ZF1

Zend Framework po stažení z internetu obsahuje všechny své základní knihovny a některé další podpůrné nástroje. Jedním z nich je shell skript umístěný v `bin/zf.sh` (pro linux) nebo `bin/df.bat` (pro windows), který za nás vytvoří základní strukturu nového projektu v ZF zadáním následujícího příkazu v shellu:<sup>11</sup>

```
zf.sh create project quickstart myproject
```

<sup>9</sup> <http://framework.zend.com/manual/1.12/en/introduction.overview.html>

<sup>10</sup> <http://framework.zend.com/about/>

<sup>11</sup> <http://framework.zend.com/manual/1.12/en/learning.quickstart.create-project.html>

Vygenerovaná adresářová struktura vypadá takto:

```
1. |-- application
2. |   |-- Bootstrap.php
3. |   |-- configs
4. |   |   `-- application.ini
5. |   |-- controllers
6. |   |   |-- ErrorController.php
7. |   |   `-- IndexController.php
8. |   |-- models
9. |   `-- views
10. |       |-- helpers
11. |       `-- scripts
12. |           |-- error
13. |           |   `-- error.phtml
14. |           `-- index
15. |               `-- index.phtml
16. |-- library
17. |-- public
18. |   |-- .htaccess
19. |   `-- index.php
20. `-- tests
21.     |-- application
22.     |   `-- bootstrap.php
23.     |-- library
24.     |   `-- bootstrap.php
25.     `-- phpunit.xml
```

Adresář *application* obsahuje zdrojové kódy samotné webové aplikace, zejména nastavení v souboru *application.ini* (připojení k databázi, nastavení modulů, namespace atd.). V souboru *Bootstrap.php* najdeme třídu *Bootstrap*, která definuje a inicializuje zdroje a komponenty použité v daném projektu. V adresářích *controllers*, *models* a *views* jsou umístěny soubory implementující architekturu model-view-controller. V MVC kontroler řídí běh programu, pro práci s daty využívá modely a předává data ke zobrazení pro view. Adresář *library* potom obsahuje knihovnu Zend Frameworku a do

adresáře *public* patří soubory, které mají být přes webový server přímo dostupné na určité URL. Sem můžeme umístit soubory s JavaScriptem a CSS styly.<sup>12</sup>

### 1.4.2 Konvence a názvosloví

Zend Framework a projekty postavené na něm dodržují několik základních konvencí.

- Názvy tříd začínají velkým písmenem a pokud jsou složeny z více slov, jsou slova oddělena pomocí tzv. CamelCase (každé nové slovo začíná velkým písmenem).
- Názvy tříd obsahují jako prefix cestu k souboru s touto třídou následující za hlavním adresářem pro MVC rozdělení, přičemž oddělovač adresářů je nahrazen za podtržítko (\_).  
Tedy třída umístěná v *controllers/Order/ProcessController.php* se musí jmenovat *Order\_ProcessController*.

### 1.4.3 Komponenty Zend Frameworku

ZF obsahuje mnoho užitečných komponent. Cílem této práce není detailně popsat všechny z nich, proto v textu abecedně shrnu jen ty nejdůležitější:<sup>13</sup>

- **Zend\_Cache** - poskytuje základní způsob pro cachování dat. Díky této komponentě je možné na základě identifikátoru získat z cache výsledek dříve provedené operace, aniž by bylo potřeba celou tuto náročnou operaci znovu provádět. Cache je vhodné použít na data, která se příliš často nemění. Také lze nastavit dobu platnosti pro každou cache.<sup>14</sup>
- **Zend\_Controller** - jedná se o základ celého model-view-controller systému v Zend Frameworku. Ten se snaží striktně oddělit logickou část aplikace od její části zobrazovací a řídí, jaký kód se bude provádět na základě dvou parametrů předaných v URL - takzvaného *kontroleru* a *akce*. Každý kontroler se v ZF potom umísťuje do separátního souboru a musí být potomkem třídy *Zend\_Controller\_Action*. Mějme ukázkovou URL adresu:  
*www.mujprojekt.cz/uzivatele\_sprava-uzivatelu/pridej-uzivatele/*

---

<sup>12</sup> <http://framework.zend.com/manual/1.12/en/learning.quickstart.create-project.html>

<sup>13</sup> <http://framework.zend.com/manual/1.12/en/reference.html>

<sup>14</sup> <http://framework.zend.com/manual/1.12/en/zend.cache.html>

Předaný název kontroleru je “uzivatele\_sprava-uzivatelu”, název akce potom “pridej-uzivatele”. ZF transformuje tyto parametry tak, že začátek každého slova převede na velké písmeno a odstraní oddělovací pomlčky. V adresářové struktuře projektu potom hledá kontroler s názvem

*Uzivatele\_SpravaUzivatelController* a na něm automaticky zavolá metodu (akci) s názvem *pridejUzivateleAction()*. Podle výše uvedených konvencí je zřejmé, že kontroler *Uzivatele\_SpravaUzivatelController* bude umístěn v souboru *controllers/Uzivatele/SpravaUzivatelController.php*.<sup>15</sup>

- **Zend\_Db** - skládá se z dalších komponent a zpřístupňuje v projektech ZF připojení k databázi, které skryje za své rozhraní, takže se programátor oprostí od konkrétního použitého databázového systému. Zend\_Db\_Adapter podporuje např. IBM DB2, MariaDB, MySQL, Microsoft SQL Server, Oracle, PostgreSQL nebo SQLite.<sup>16</sup>
- **Zend\_Debug** - poskytuje ladící funkce pro snadnější ladění programu.
- **Zend\_Form** - umožňuje definovat, vykreslovat a dále pracovat s formulářovými prvky webové aplikace. Stará se o validaci údajů ve formuláři, předvyplňování defaultních hodnot a plní další užitečné úkoly.
- **Zend\_Layout** - umožňuje zobrazovací šablony organizovat do větších strukturovaných celků.
- **Zend\_Locale** - pracuje s nastavením správného jazyka, formátu data a času, měny a dalších parametrů souvisejících s lokalizací aplikace.
- **Zend\_Log** - komponenta pro elegantní práci s chybovými logy aplikace.
- **Zend\_Mail** - zabezpečuje práci s elektronickou poštou (odesílání a čtení).
- **Zend\_Pdf** - komponenta umožňující generování dokumentů ve formátu PDF.
- **Zend\_Session** - pro práci se session (sezením).
- **Zend\_Soap** - umožňuje práci s webovými službami.
- **Zend\_Translate** - překlad samotného obsahu webové aplikace. Využívá se pro multijazyčné aplikace.

---

<sup>15</sup> <http://framework.zend.com/manual/1.12/en/zend.controller.quickstart.html>

<sup>16</sup> <http://framework.zend.com/manual/1.12/en/zend.db.adapter.html>

- **Zend\_View** - třída pro práci s view částí systému model-view-controller. Má tedy na starosti zobrazování, vykreslování struktury a dat v aplikaci. Data, která potřebuje mít view dostupná pro zobrazení jí předává kontroler.

## 1.5 MySQL

Jedná se o výkonný relační databázový systém, který běží na serverech a nejčastěji je využíván pro webové aplikace. MySQL je ideální pro využití jak na malých projektech, tak na rozsáhlých webových aplikacích. Přestože má jako každá technologie své odpůrce, jedná se o spolehlivý, velice rychlý databázový systém, který se snadno používá. Je kompilován pro mnoho platform, může tedy běžet jak na linuxovém serveru, tak na operačním systému Windows nebo OS X. MySQL je aktuálně vyvíjena a podporována společností Oracle, mezi jejíž produkty mimo jiné patří i konkurenční komerční databázový systém Oracle Database.<sup>17</sup>

Když firma Oracle koupila v roce 2009 společnost Sun Microsystems a s ní i MySQL, vzniklo několik edicí MySQL. Kromě svých komerčních verzí Oracle udržuje v nabídce i tzv. komunitní edici, která je založena na open source komunitním vývoji a je dostupná zdarma pod licencí GPL.<sup>18</sup> Z důvodu obav o zachování otevřenosti MySQL po jejím přechodu pod Oracle v roce 2012 založili původní autoři MySQL organizaci MariaDB Foundation, která dále udržuje a vyvíjí původní open source MySQL projekt pod názvem MariaDB. Ta je plně kompatibilní s MySQL, takže přechod na ni neskýtá žádné riziko.<sup>19</sup>

## 1.6 Webové služby

Webovými službami nazýváme systém, který umožňuje vzájemnou spolupráci dvou systémů či strojů v počítačové síti. Existují různé implementace webových služeb, ale v rámci této práce se zaměřím na webové služby, které jsou popsány jazykem WSDL (Web Services Description Language) a komunikace probíhá pomocí SOAP protokolu. Jak WSDL, tak i SOAP jsou založeny na formátu XML. Služby implementované na základě těchto standardů jsou nezávislé na implementaci a dají se provozovat na

<sup>17</sup> [http://www.w3schools.com/php/php\\_mysql\\_intro.asp](http://www.w3schools.com/php/php_mysql_intro.asp)

<sup>18</sup> <http://www.mysql.com/products/community/>

<sup>19</sup> <http://www.root.cz/clanky/databaze-mariadb-valcuje-mysql/>

různých platformách. Mohou spolu díky tomu komunikovat systémy implementované v různých programovacích jazycích.

### **1.6.1 SOAP protokol**

Název protokolu SOAP byl původně zkratkou pro Simple Object Access Protocol. Vzhledem k tomu, že hlavním účelem specifikace protokolu SOAP je předávat standardní formát zpráv, od verze 1.2 je slovo SOAP považováno za samostatný termín.

#### **Historie**

Protokol byl navržen v roce 1998 pány jménem Dave Winer, Don Box, Bob Atkinson a Mohsen Al-Ghosein. Jedná se o novějšího nástupce staršího protokolu RPC. Od něj některé vlastnosti přejímá, ale snaží se jej rozšířit.

Na protokolu SOAP staví svou funkcionalitu webové služby, díky kterým je možné realizovat spolupráci mezi systémy v síti. Tyto systémy si mohou vzájemně poskytovat funkcionalitu, popřípadě předávat data. Komunikace webových služeb je založena na zprávách. Tyto zprávy bylo potřeba standardizovat tak, aby používaly stejný formát bez ohledu na původ systému, který danou zprávu zasílá.<sup>20</sup>

#### **Použití**

SOAP protokol může pro zprávy používat různé šablony, z nichž nejznámější je RPC šablona. Ta je charakteristická tím, že komunikace probíhá mezi klientem a serverem. V tomto případě server poskytuje určitou funkcionalitu, kterou může klient využívat formou volání různých metod na webové službě.

Zpráva protokolu SOAP je obvykle přenášena pomocí jiných, zavedených protokolů. Může to být buď SMTP nebo HTTP protokol. Dnes je však díky své rozšířenosti pro tyto účely používán protokol HTTP. Standardem pro přenos SOAP zprávy se potom stal formát XML.

SOAP byl zvolen jako protokol, který využívají webové služby, zejména z důvodu, že umožňuje přenos zpráv, které jsou inteligentní a maximálně samostatné. Jeho výhodou je tedy možnost široké rozšířitelnosti.<sup>21</sup>

---

<sup>20</sup> <http://www.w3.org/TR/soap12-part1/>

<sup>21</sup> <http://www.w3.org/TR/soap12-part1/>

Jak jsem již zmínil, protokol je postaven na syntaxi formátu XML. Jedná se o textový dokument, který se skládá ze struktury elementů, jejich atributů a obsahu.. Příklad útržku obecného dokumentu v XML může vypadat např. takto:

Struktura:

```
<orders>
  <order id="12345" vs="4443344" totalPrice="1500" currency="Kč">
    <item>
      <name>Servisní práce</name>
      <price>1500</price>
      <currency>Kč</currency>
    </item>
  </order>
</orders>
```

Zprávu SOAP tvoří takzvaná obálka, hlavička a samotné tělo zprávy. Strukturu znázorňuje obrázek 1.



Obrázek č. 1: Základní struktura zprávy SOAP. (Zdroj: Vlastní zpracování)

### Obálka, hlavička a tělo zprávy

Do obálky je každá zpráva zabalená a tím jsou uchovány všechny její části pohromadě. Hlavička je v SOAP protokolu sice nepovinná, ale je ve většině případů uvedena, jelikož je pro efektivní využití celého protokolu velice důležitá. Umožňuje totiž použití jeho rozšíření.

Hlavička bývá rozdělena do jednotlivých bloků – takzvané bloky hlaviček. Ty implementují nezávislost zpráv a díky nim mohou zprávy obsahovat množství různých



pomocných informací. Například informace jak obsah doručit, jak jej zpracovat apod. Díky blokům hlaviček je systém webových služeb rozšiřitelný a komponovatelný.<sup>22</sup>

Následuje tělo samotné zprávy, což jsou v podstatě data formátována ve formátu XML.

Níže uvádím příklad, jak by mohl vypadat dotaz klienta na server a následná odpověď serveru. Ukázka dotazu SOAP klienta na informace o produktu:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getProductDetails xmlns="http://shop.example.com/ws">
      <productID>987545</productID>
    </getProductDetails>
  </soap:Body>
</soap:Envelope>
```

Odpověď webové služby pro klienta:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getProductDetailsResponse xmlns="http://shop.example.com/ws">
      <getProductDetailsResult>
        <productName>Zmrzlina banánová</productName>
        <productID>987545</productID>
        <popis>Smetanová zmrzlina s příchutí banánu</popis>
        <cena>38</cena>
        <naSkladu>ano</naSkladu>
      </getProductDetailsResult>
    </getProductDetailsResponse>
  </soap:Body>
</soap:Envelope>
```

### Přílohy a chyby

Přílohy jsou určeny pro přenos dat, která nelze jednoduše formátovat do XML. Mezi tato data patří binární data - například obrázky.

Součástí zprávy může být také popis chyb. Je to nepovinná část, která může obsahovat jednoduchou chybovou zprávu. Ta je potom použita v případě selhání nebo při obsluze výjimek.<sup>23</sup>

---

<sup>22</sup> ERL, Thomas. SOA: servisně orientovaná architektura, Brno, 2009, s.125-127

<sup>23</sup> ERL, Thomas. SOA: servisně orientovaná architektura, Brno, 2009, s.128-129

## Uzly

Uzly jsou programy jejichž služby přijímají nebo odesílají SOAP zprávy. Jednotlivá zpráva může procházet přes několik takových uzlů. Každá komunikace obsahuje počátečního odesílatele SOAP zprávy a konečného příjemce SOAP zprávy. Mezi nimi mohou být i takzvaní prostředníci SOAP, přes které přechází zpráva mezi odesílatelem a příjemcem. Prostředník SOAP je tedy aktivní odesílatel, směřuje SOAP zprávu k cíli a při průchodu většinou modifikuje hlavičku této zprávy.<sup>24</sup>

Mezi uzly tedy řadíme:

- odesílatel SOAP
- příjemce SOAP
- prostředník SOAP
- počáteční odesílatel SOAP
- konečný příjemce SOAP

### 1.6.2 WSDL

Jazyk WSDL je také založen na XML formátu a používá se pro samotný popis webových služeb. Popisné dokumenty musí doprovázet každou webovou službu, která chce vystupovat jako konečný příjemce, aby mohla být tato komunikace vedena konzistentně, avšak volně vázanou formou. Popis služeb lze rozdělit do dvou kategorií: abstraktní popis a konkrétní popis. Abstraktní popis deklaruje vlastnosti rozhraní webové služby, aniž by uváděl technologie použité pro přenos. Díky tomu, že je zde tato informace oddělená od technologické platformy pozadí služby, je zajištěna integrita tohoto popisu služby. Konkrétní popis potom obsahuje popis webové služby rozhraní v souvislosti s konkrétní technologií. Skládá se ze 3 částí: binding, port a service.<sup>25</sup>

## 1.7 Návrhové vzory

První návrhové vzory poprvé představila skupina GoF (Gang of Four) na začátku devadesátých let minulého století. Jedná se o vzorové postupy řešení často se vyskytujících programátorských problémů. Pojem návrhové vzory je velice úzce spjat

---

<sup>24</sup> ERL, Thomas. SOA: servisně orientovaná architektura, Brno, 2009, s.129-133

<sup>25</sup> ERL, Thomas. SOA: servisně orientovaná architektura, Brno, 2009, s.116-120

s takzvaným objektovým programováním, které se stalo velice oblíbeným a moderním při dnešním vývoji software. Existuje několik málo desítek návrhových vzorů, přičemž každý z nich se snaží co nejefektivněji (s využitím takzvaných best practices) řešit určitý konkrétní typový úkol. Návrhové vzory jakožto postupy řešení nejsou závislé na konkrétním použitém programovacím jazyce. Jedná se o obecné postupy. Pro programátory mají mnoho pozitivních faktorů, zejména zrychlují řešení a zjednodušují a zkvalitňují komunikaci. Mezi návrhové vzory, které využívá při vývoji webových aplikací například výše popsany Zend Framework, patří například Jedináček (singleton), Továrna (Faktory), Iterator, Adaptér a Model-view-controller.<sup>26</sup>

## 1.8 HTML

HTML je zkratkou pro *HyperText Markup Language*. Jedná se o značkovací jazyk podobný jazyku XML. Dokument HTML se skládá z hierarchické struktury takzvaných elementů, jejichž názvy jsou uzavřeny do ostrých závorek (<, >). Elementy mohou mít svůj obsah a své atributy, jejichž hodnota se uzavírá do uvozovek. Jazyk HTML se používá zejména pro vytváření webových stránek, dokáže definovat strukturu takovéto stránky, její obsah a informace o jejím zobrazení pro webový prohlížeč. Samotný HTML dokument je v dnešní době většinou dynamicky generován některým ze serverových programovacích jazyků - slouží jako výstup pro uživatele.

Příklad HTML elementu *DIV* s textovým obsahem a atributy *id*, *class* a *data-pocet-znaku*:

```
<div id="unikatni-nazev-elementu" class="trida-pro-css-definici" data-pocet-znaku="23">Toto je obsah elementu</div>
```

HTML vzniklo začátkem devadesátých let minulého století, v roce 1997 byla vydána verze 4, v roce 1999 potom verze 4.01, následována takzvanou XHTML verzí. Dlouhou dobu byla vyvíjena verze HTML5, která dostala svou finální specifikaci teprve v roce 2014. Mezi novinky HTML5 patří zkrácená specifikace typu dokumentu (<!DOCTYPE html>). HTML5 také obsahuje nové elementy pro definici struktury dokumentu jako například <section>, <article>, <header>, <footer>. Důležitou novinkou je podpora multimédií s elementy <video> a <audio>, nebo také element <canvas>, který umožňuje použít ve stránce vektorovou grafiku. Mezi další významná vylepšení patří

---

<sup>26</sup> PECINOVSKÝ, Rudolf. Návrhové vzory, 2007, s.34-37

nové typy formulářových elementů (*<input>*), které v předchozích verzích již dlouho chyběly a jejichž funkcionalita byla často implementována pouze díky JavaScriptu. Jedná se o typ *date* (pro zadávání vstupu data), typ *URL* (pro zadávání internetové adresy), typ *email* (pro zadávání e-mailové adresy) nebo například typ *color* (pro možnost výběru barvy z barevné palety). Další významnou vlastností HTML5 je také možnost použití lokálního úložiště (*localStorage*) přímo v prohlížeči (bez nutnosti připojení k serveru).<sup>27</sup>

Vývoj jazyka HTML podobně jako ostatních webových technologií (CSS, JavaScript) do velké míry ovlivňují autoři webových prohlížečů, kteří různé vlastnosti tohoto jazyka podporují, popř. vypouštějí nebo rozšiřují. Významným hráčem je rovněž organizace *W3C* (World Wide Web Consortium), která spolu s veřejností vyvíjí standardy a specifikaci pro tento jazyk. W3C netvoří standardy pouze pro jazyk HTML, ale také pro další webové technologie, např. SOAP, WSDL, XML, DOM atd.

## 1.9 CSS

CSS neboli kaskádové styly jsou jazykem, který definuje vzhled a zobrazení HTML dokumentu (webové stránky). Hlavním důvodem jeho vzniku bylo oddělení způsobu zobrazení od samotného obsahu stránky. Tím je možno vzhled upravit, aniž by programátor musel zasahovat do samotného HTML kódu. Definici kaskádových stylů lze zahrnout do samotného HTML kódu jako obsah elementu *<style></style>*, ale tento způsob se většinou nevyužívá a ani nedoporučuje. Nejčastěji je totiž definice CSS extrahována do samostatných souborů s příponou *.css*, na které jsou v samotném HTML dokumentu uvedeny pouze reference. Mezi výhody separace CSS kódu do zvláštního souboru patří například větší přehlednost samotného HTML kódu, ale také snížení přenášených dat směrem ke klientovi (do prohlížeče), neboť moderní internetové prohlížeče si tyto CSS soubory ukládají do mezipaměti (cache) a nestahují je při každém požadavku ze serveru. Syntaxe kaskádových stylů je poměrně jednoduchá. Definice začíná tzv. selektorem, který určuje, na které HTML elementy se mají aplikovat vlastnosti uvedené v samotné definici. Definice aplikovaných vlastností se potom uzavírá do složených závorek. Nejzákladnějšími selektory jsou názvy samotných elementů, které je možno upřesnit názvem atributu - např. *id* (v css znak "mřížka" - #)

---

<sup>27</sup> KEITH, Jeremy. HTML5 FOR WEB DESIGNERS, New York, 2010, s.24-70

nebo *class* (v css znak “tečka”). Existuje však celá řada dalších, pokročilých selektorů. Komentáře v CSS je možno psát mezi */\** a *\*/*.

Příklad:

```
/* Nastaví barvu pozadí na zelenou a barvu písma na bílou u elementu <div>
s atributem id="hlavicka". Tedy:
```

```
<div id="hlavicka"></div>
```

```
*/
```

```
div#hlavicka { background-color: green; color: white; }
```

```
/* Nastaví tučné zobrazení textu u všech elementů <a> (hypertextový odkaz),
které mají atribut class="aktivni" a je zanořen do elementu <div> s atributem
id="menu". Tedy:
```

```
<div id="menu">
```

```
<a href="http://www.w3c.com" class="aktivni">odkaz</a>
```

```
<a href="http://www.example.com" >example</a>
```

```
*/
```

```
div#menu a.aktivni { font-weight: bold; }
```

Podrobný popis všech vlastností a jejich hodnot, které lze pro konkrétní HTML element použít jsou k dispozici například na webu W3C.<sup>28</sup>

## 1.10 JavaScript

Je velice rozšířený, hojně používaný skriptovací jazyk, který patří do kategorie interpretovaných jazyků (jeho interpretem je v drtivé většině případů internetový prohlížeč). Jedná se o objektově orientovaný programovací jazyk, takže při psaní JavaScriptových programů využíváme zásady OOP. Základní podpora JavaScriptu je obsažena ve všech moderních internetových prohlížečích. Některé z nich (například Google Chrome) na něm staví i svá vlastní rozšíření či dialogy pro nastavení prohlížeče samotného. JavaScript je rovněž součástí konceptu HTML5 a díky němu je možno v prohlížeči implementovat rozsáhlé webové aplikace, které v dnešní době již dokáží nahradit některé aplikace desktopové. JavaScriptové programy si tedy nejdříve prohlížeč stáhne z webové adresy do počítače a následně je spouští až na straně uživatele. Stejně jako u kaskádových stylů je možno zdrojový kód vkládat přímo do

---

<sup>28</sup> <http://www.w3schools.com/css/default.asp>

HTML stránky (jako obsah elementu `<script>`), ale upřednostňovaný způsob je separace javascriptového kódu do externích souborů s příponou `.js`, které jsou opět potom do HTML kódu vkládány pouze jako reference na tento soubor.

Standardizovaná verze JavaScriptu nese název ECMAScript (podle asociace, která tento standard sestavila). Pravdou je, že každý internetový prohlížeč má svou implementaci JavaScriptu, ve které některé jeho vlastnosti může vynechat a naopak přichází s vlastnostmi vlastními. Rozdíly, které vznikají při interpretaci JavaScriptu v různých prohlížečích, může při vývoji sjednotit použití různých javascriptových frameworků, které se s různou implementací JavaScriptu vypořádají za programátora. Díky JavaScriptu je možno ve webové stránce nebo aplikaci realizovat různé grafické animace, a také poskytnout interaktivní chování celé aplikaci. Interaktivitu lze zajistit zejména technologií AJAX (*Asynchronous JavaScript and XML*), která umožňuje pomocí JavaScriptu zaslat HTTP požadavek na server a získat z něj zpět data, aniž by bylo nutné znovu načíst celou zobrazenou stránku. Takto získaná data je možno opět pomocí javascriptu interaktivně zpracovat do aktuálně zobrazeného HTML a stránka se tak dynamicky změní. S technologií AJAX se setkáváme pod tímto názvem od roku 2005<sup>29</sup> a od té doby je hojně využívána při tvorbě takzvaného webu 2.0 - dnes v HTML5.

## 1.11 jQUERY

jQuery je bohatá javascriptová knihovna (framework), která je malá, rychlá a velice snadno se používá a integruje do webové aplikace. Umožňuje manipulaci s HTML dokumentem za běhu stránky, elegantně zpracovávat zachytávání událostí v prohlížeči (kliknutí na HTML element, přejetí myši, stisk klávesy na klávesnici), provádět animace a také velice jednoduše používat technologii AJAX. jQuery zaručuje chod svých funkcí přes všechny implementace JavaScriptu u různých prohlížečů. Pro tento framework existuje celá řada pluginů a rozšíření, které umožňují programovat interaktivní webovou aplikaci. Na frameworku jQuery je postavené takzvané jQuery UI (User Interface), což je knihovna různých funkcí a elementů uživatelského rozhraní jako jsou grafické efekty, přesouvání elementů ve stránce, řazení elementů systémem Drag&Drop (přesunutím myši), zobrazení interaktivních dialogových oken, záložek, tooltipů, menu a velkou

---

<sup>29</sup> <https://www.interval.cz/clanky/ajax-a-php-tvorime-interaktivni-webove-aplikace-profesionalne/>

spoustou dalších funkcí. Navíc může kdokoli napsat svůj vlastní, neoficiální plugin pro knihovnu jQuery a dát ji ostatním k dispozici na internetu. Poslední verze jQuery nese označení 2.1.4.<sup>30</sup>

## 1.12 GIT

Jedná se o volně použitelný, open source verzovací systém, který je navrhnut tak, aby dokázal spravovat vše od malých projektů až po velice rozsáhlé aplikace a to s maximální rychlostí a efektivitou. Autorem verzovacího systému GIT je Linus Torvalds, autor linuxového jádra, pro něhož se staly ostatní soudobé verzovací systémy těžko použitelné a neefektivní při vývoji tak rozsáhlého projektu, jakým linuxové jádro je. Proto se rozhodl pro vývoj GITu. Ten na rozdíl třeba od staršího systému Subversion není centralizovaný, ale se jedná o distribuovaný verzovací systém. To znamená, že každý programátor, který si z tzv. repozitáře zdrojových kódů dané aplikace stáhne svou pracovní kopii, má lokálně k dispozici veškerou historii změn zdrojového kódu bez nutnosti další komunikace se serverem. GIT umožňuje efektivně pracovat s různými vývojovými větvemi, které se využívají pro vývoj nových funkcí. Má excelentní možnosti pro takzvané “mergování” zdrojového kódu neboli zahrnování různých vývojových věcí do jiné (hlavní) vývojové větve. Díky rozsáhlé komunitě existuje kvalitní dokumentace tohoto nástroje. Je multiplatformní a tedy dostupný jak pro operační systémy Windows, Mac OS, tak samozřejmě pro Linux. Standardně se používá v takzvaném příkazovém řádku, ale existuje pro něj řada grafických nadstavb. Mezi společnostmi a projekty, které využívají při vývoji softwaru GIT patří například Google, Facebook, Microsoft, Twitter, LinkedIn nebo je v něm verzován mobilní operační systém Android.<sup>31</sup>

---

<sup>30</sup> <https://jquery.com/>

<sup>31</sup> <https://git-scm.herokuapp.com/>

## **2 Analýza současného stavu**

Než se začnu věnovat návrhu řešení nového nabídkového systému, je potřeba se zastavit nad aktuálním stavem, který předchází novému nabídkovému systému.

### **2.1 Popis firmy**

Firma je online cestovní agenturou, která prodává zájezdy od více než 300 českých, slovenských a německých cestovních kanceláří, a to za stejné ceny jako samotné cestovní kanceláře. Na trhu působí již více než 12 let a za tu dobu se stala jedničkou v online prodeji tohoto druhu. Její obrat každoročně roste a své aktivity rozšiřuje do dalších zemí střední a východní Evropy. Zákazníkům kromě největšího výběru zájezdů nabízí i cestovní pojištění, letenky a další produkty či služby. To vše s kvalitním zákaznickým servisem. Činnost firmy stojí na prodeji a zejména na technologickém zázemí, které samotný prodej umožňuje a zefektivňuje prodejní i zákaznický proces. Firma má rozsáhlou síť franchisových poboček, kterých je po celé ČR již více než 100. Společnost je držitelem certifikátu systému managementu kvality ISO 9001.



### **Struktura společnosti:**

Ve společnosti existují následující oddělení, která zajišťují jednotlivé činnosti:

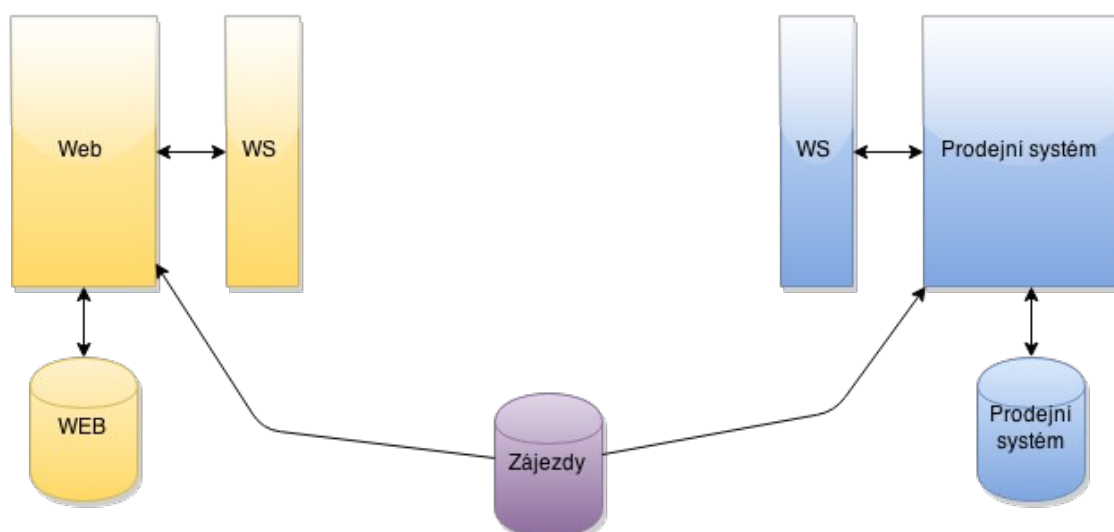
- Vedení firmy
- Marketingové oddělení
- Produktové oddělení
- Oddělení pokynů a odbavení
- Oddělení recepčních
- Prodejní oddělení - call centrum
- Finanční oddělení
- Personální oddělení
- IT oddělení
- Vývojové oddělení
- Oddělení Data Entry

Společnost nabízí následující služby a produkty:

- Katalogové zájezdy
- Last Minute zájezdy
- Zájezdy německých CK
- First Minute zájezdy
- Ubytování v ČR i zahraničí
- Recenze hotelů
- Fotogalerie
- Atlas světa - informace o zemích
- Dárkový certifikát
- Letenky do celého světa
- Affiliate (partnerský) program
- Franchising

## Hlavní informační systémy společnosti a jejich vazby na databáze:

Mezi dva hlavní systémy, které společnost využívá při prodeji patří webový portál (provozovaný pro několik zemí na různých doménách) a interní prodejní systém. Ten implementuje a zabezpečuje stěžejní obchodní logiku celého prodejního procesu a využívají jej prodejci, call centrum i franchisoví partneři pro práci s objednávkami a klienty.



Obrázek č. 2: Schéma hlavních systémů použitých ve společnosti. (Zdroj: Vlastní zpracování)

Oba zmíněné systémy jsou provozovány jako webové aplikace. Využívají tedy tzv. třívrstvou architekturu, ve které jsou odděleny datová, aplikační a prezentační část. Data mezi sebou si vyměňují vždy pouze dvě sousední vrstvy. Třívrstvou architekturu znázorňuje následující obrázek.



Obrázek č. 3: Schéma třívrstvé architektury webové aplikace. (Zdroj: Vlastní zpracování)

Ve skutečnosti je však v tomto případě aplikační vrstva u obou systémů tvořena několika webservery, mezi které rozděluje požadavky další server - tzv. load balancer. Díky tomu je možno distribuovat zátěž rovnoměrně na více serverů a tím posílit výkon.

## **2.2 Popis procesů**

Z firemních procesů nás bude v rámci této práce zajímat zejména prodejní proces.

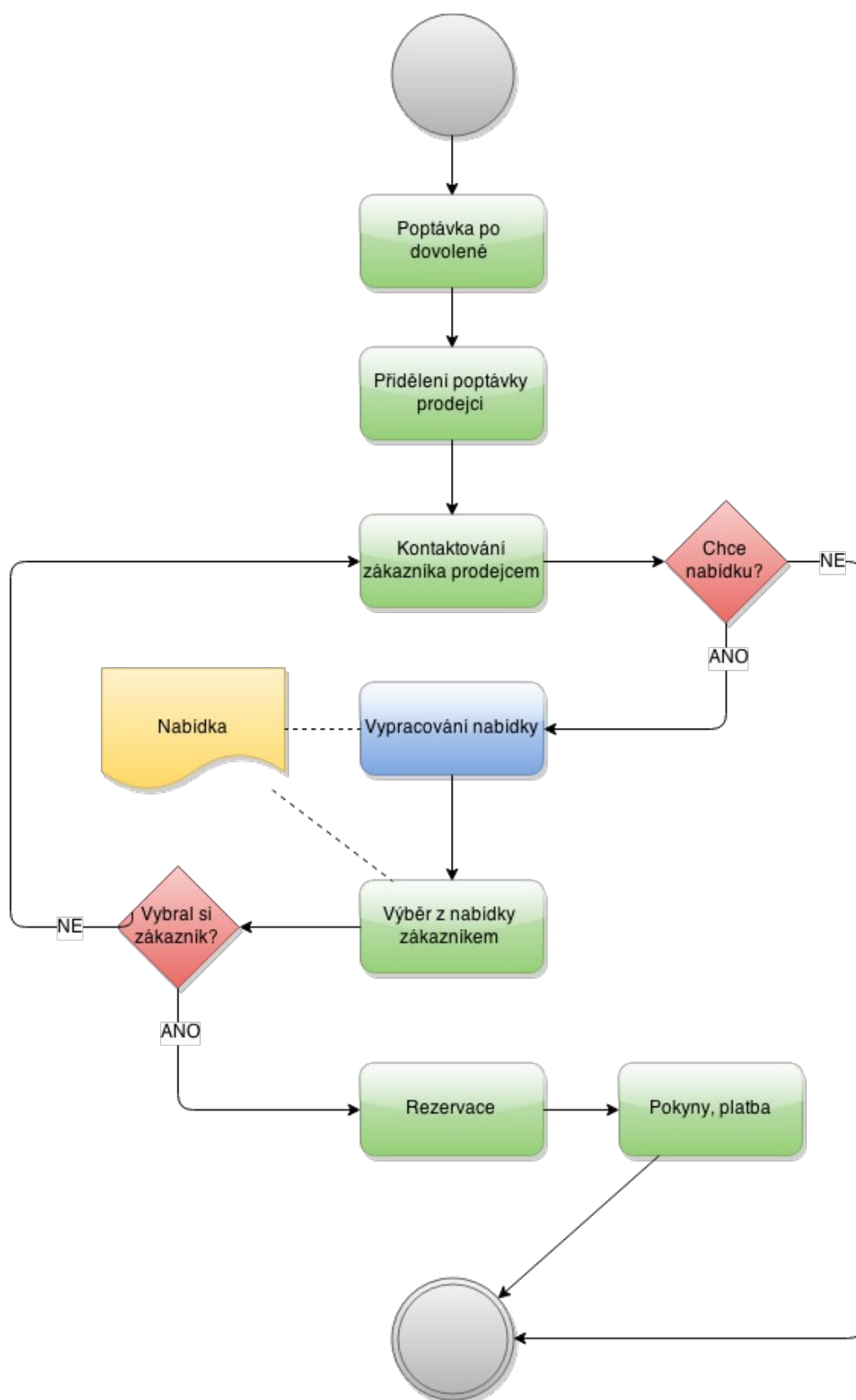
### **2.2.1 Prodejní proces**

Prodejní proces začíná poptávkou po dovolené ze strany konkrétního zákazníka. Ta může přijít několika způsoby:

- Vyplněním formuláře s nezávaznou objednávkou na portálu společnosti, kde jsou zveřejněny zájezdy v nabídce.
- Telefonátem do call centra na bezplatnou linku společnosti a zadáním nezávazné objednávky do prodejního systému prodejcem.
- Tzv. zpětným telefonátem, kdy zákazník na sebe na webu zanechá telefonní číslo, na které si přeje být ohledně zájezdu kontaktován.
- Návštěvou zákazníka na některé z kamenných poboček, kde sdělí své požadavky prodejci.

Samotný proces tedy začíná zákaznickovou poptávkou, tzv. nezávaznou objednávkou, která je přiřazena konkrétnímu prodejci. Ten zákazníka zpětně kontaktuje pro upřesnění jeho požadavků na dovolenou. Na základě těchto požadavků z dostupných zájezdů sestaví zákazníkovi nabídku s několika variantami. Může se také stát, že zákazník má již dopředu vybrán konkrétní zájezd z nabídky na webovém portálu a další alternativní nabídky již nepožaduje. V tomto případě dojde pouze k upřesnění informací a realizaci závazné objednávky. Po konečném ověření ceny a závazném výběru konkrétního zájezdu potom prodejce připraví zákazníkovi smlouvu, dohodne se na platbě a dalším postupu. Až do odjezdu je mu v případě potřeby k dispozici.

Jednotlivé základní kroky prodejního procesu popisuje následující diagram.



Obrázek č. 4: Vývojový diagram prodejního procesu. (Zdroj: Vlastní zpracování)

Je vidět, že jednou z aktivit, které prodejce často a opakovaně provádí je vytváření nabídky na míru zákazníkovi. Proto je vhodné tento proces co nejvíce optimalizovat, aby došlo ke snížení časových nároků na tvorbu nabídky, minimalizovala se pravděpodobnost výskytu chyb a v neposlední řadě, aby nabídky byly zákazníkovi předkládány v přehledném a jednotném firemním stylu a formě.

Při tvorbě nabídky musí prodejce vykonat několik kroků:

- Nalezení vhodných zájezdů podle požadavků zákazníka (destinace, počet osob, cena, doprava, strava, vzdálenost od pláže, podmínky pro malé děti atd.).
- Doplnění vybraných zájezdů o cenovou kalkulaci, která vychází z cen, které jsou k dispozici z dat od CK (cestovní kanceláře). Výpočet orientační celkové ceny ke každé variantě nabídky.
- Zkopírovat vybrané nabídky, informace k nim včetně fotografie do e-mailu pro klienta, popř. do dokumentu pro tisk.
- Zaslat nabídku zákazníkovi.
- Zaznačit tuto nabídku do prodejního systému.
- Kontaktovat zákazníka s doplňujícími informacemi a dotazem na jeho preferenci.
- Zaznačit nabídku, kterou si zákazník vybral do prodejního systému.

Redukce a automatizace těchto kroků je právě úkolem nabídkového systému, který je předmětem této práce.

## 2.3 SWOT analýza

Jedná se o techniku analýzy stavu firmy, která se používá pro strategické rozhodování a řízení. Jde o zkratku z anglických slov Strengths (silné stránky), Weaknesses (slabé stránky), Opportunities (příležitosti) a Threats (hrozby). Zatímco silné a slabé stránky popisují zejména interní prostředí podniku, příležitosti a hrozby se zaměřují na vnější prostředí podniku.

**Tabulka č. 2: SWOT analýza podniku.** (Zdroj: Vlastní zpracování)

<b>S - Silné stránky</b> <ul style="list-style-type: none"><li>• partnerská síť</li><li>• široký dosah na zákazníky díky internetu</li><li>• komplexnost služeb (zájezd, ubytování, letenky, pojištění)</li><li>• obrovská nabídka</li><li>• profesionální, mladý a inovativní tým</li><li>• technologické zázemí</li></ul>	<b>W - Slabé stránky</b> <ul style="list-style-type: none"><li>• sezónnost, fluktuace brigádníků</li><li>• nelze vyloučit lidské selhání, nedokonalé proškolení</li><li>• nepřesné informace na webu</li><li>• nedodržení standardu firmy na pobočce</li></ul>
<b>O - Příležitosti</b> <ul style="list-style-type: none"><li>• rozšíření nabídky o další služby</li><li>• rozšíření a posílení postavení na nových trzích</li><li>• využití nových technologií, které se stávají oblíbené mezi zákazníky</li></ul>	<b>T - Hrozby</b> <ul style="list-style-type: none"><li>• nespokojení klienti</li><li>• nákupy zájezdů či služeb u jiného zprostředkovatele (dodavatele)</li><li>• finanční krize, zvýšení ceny Eura</li></ul>

### Silné stránky:

- Partnerská síť: společnost má širokou síť partnerů zejména v ČR, ale také na Slovensku, v Maďarsku, Polsku nebo v Rusku. Mezi partnery patří např. cestovní kanceláře, s nimiž si vyměňují data a jejichž zájezdy prodává, dále potom franchisové pobočky a další společnosti, jejichž produkty či služby nabízí.
- Široký dosah na zákazníky: díky tomu, že společnost zveřejňuje nabídku zájezdů, včetně potřebných informací na internetu i s možností pokročilé komunikace a nezávazné objednávky přímo přes web, maximalizuje se tím počet možných zákazníků. Díky značné rozšířenosti internetu si může kdokoli

v pohodlí svého domova vybrat z největší nabídky na internetu, porovnat zájezdy podle recenzí dřívějších klientů apod. Pro ty, kteří preferují osobní jednání nebo telefonickou objednávku existují i tyto možnosti.

- Obrovská nabídka: díky spolupráci s více než 300 cestovními kancelářemi a pravidelné aktualizací jejich nabídky může společnost nabídnout zákazníkům 3,5 mil. aktivních termínů zájezdů.
- Profesionální, mladý a inovativní tým: společnost klade důraz na kvalitní výběr vhodných zaměstnanců a na jejich důkladné a dlouhodobé školení. Celkově se jedná o mladý tým odborníků, kteří přesně vědí, jakou roli ve společnosti mají.
- Technologické zázemí: jelikož se jedná o online cestovní agenturu, její služby stojí především na využití ICT (informační a komunikační technologie). Své systémy firma neustále rozvíjí, přizpůsobuje požadavkům prodeje i zákazníků, udržuje, zabezpečuje, monitoruje a aktualizuje data.

#### **Slabé stránky:**

- Sezónnost, fluktuace brigádníků: část zaměstnanců posiluje firemní tým pouze na sezónu, která spadá do období duben - září. Někteří tito brigádníci jsou však např. studenti, kteří takto posilují tým každoročně a případně po dokončení studia nastoupí celoročně.
- Lidské selhání: tuto variantu nelze nikdy zcela vyloučit, ale společnost se jí snaží předcházet důkladným výběrem kandidátů a zejména kvalitním vstupním školicím systémem.
- Nepřesné informace na webu: díky obrovskému množství informací, které jsou k zájezdům na webu, se může stát, že některá informace není plně aktuální a je třeba ji ověřit či upřesnit. Také se může stát, že zkrácená informace je dodána nechtěně přímo cestovní kanceláří. Tomuto mohou pomoci předcházet výkonnější technologie automatického zpracování a kontrola.
- Nedodržení standardu na pobočce: může být těžší dohlížet na dodržování firemních procesů a standardů na vzdálenějších, menších pobočkách oproti hlavním pobočkám či centrále. To může poškodit jméno firmy. Zamezit tomu pomáhá využívání firemního prodejního systému. Důležité je také pravidelné školení pracovníků a vhodně postavené smluvní podmínky.

**Příležitosti:**

- Rozšíření nabídky o další služby: jelikož má firma širokou základnu zákazníků, může zkoumat jejich požadavky a vhodně jim přizpůsobit i nabídku služeb.
- Rozšíření a posílení postavení na nových trzích: firma rozšiřuje své působení na nových trzích ve střední a východní Evropě. Tam společnost vstupuje postupně a je zde prostor pro rozvoj a posílení postavení.
- Využití nových technologií, které se stávají oblíbené mezi zákazníky: v době, kdy jsou neustále vyvíjeny nové technologie a zákazníci je rádi používají, existuje mnoho možností, jak klienty zaujmout a přitáhnout.

**Hrozby:**

- Nespokojení klienti: nespokojenost klientů je hrozbou každého podniku. Může pramenit z výpadků služeb, špatné komunikace nebo nedodržení smlouvaného termínu. Někdy se však stává, že nespokojený klient u cestovní agentury reklamuje služby, které agentura nemůže ovlivnit (např. obsluha v hotelu, špatné počasí, atd). V těchto případech je nutné zákazníkovi vysvětlit, že existují skutečnosti, které firma nemůže ovlivnit.
- Nákupy zájezdů či služeb u jiného zprostředkovatele (dodavatele): může souviset s předchozím bodem. Nespokojenost klientů by mohla vést k jejich sníženému zájmu o nákup. Silné a stabilní postavení na trhu podpořené neustálou inovací technologií a systémů podporuje tuto hrozbu eliminovat.
- Finanční krize, zvýšení ceny Eura: v případě situace, kdy domácnosti budou mít napjatý rodinný rozpočet v důsledku finanční krize či jiných ekonomických vlivů, je předpoklad, že obecně omezí letní dovolené na úkor potřebnějších nákupů. Podobně při zvýšení ceny Eura se stane dovolená v zahraničí pro zákazníky dražší a může je to negativně ovlivnit při rozhodování, zda zakoupit zájezd či ne.



### 3 Návrh řešení

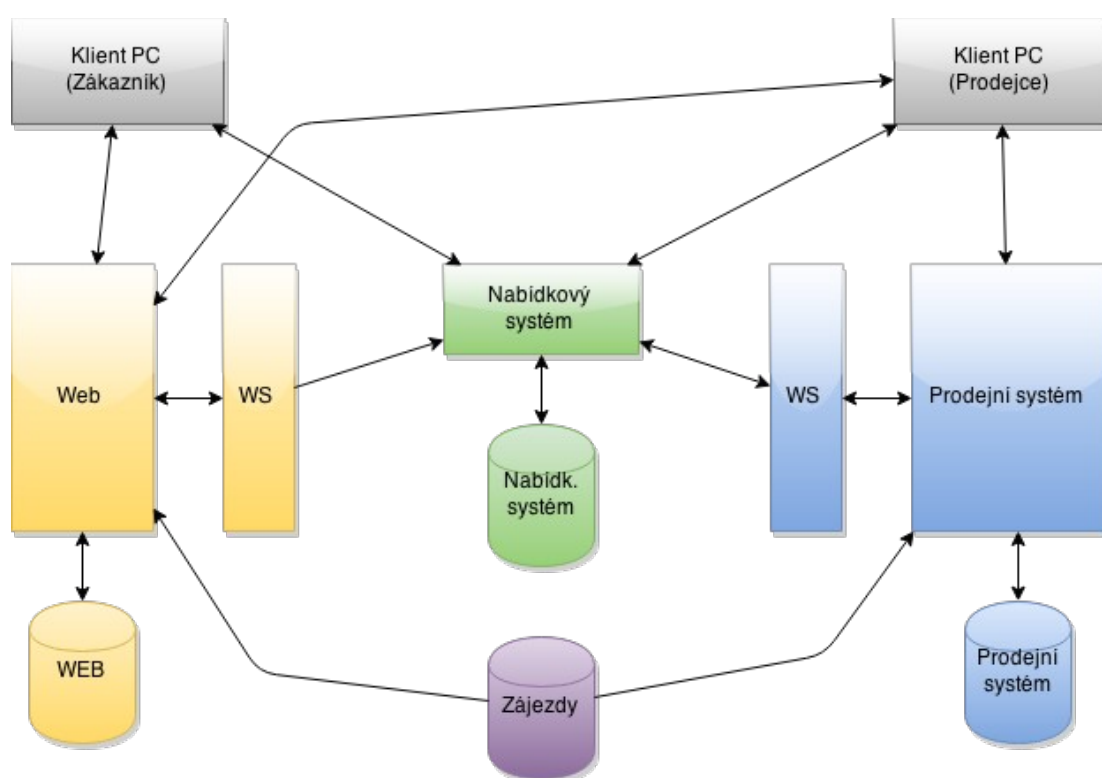
Při návrhu a implementaci nabídkového systému byla potřeba kromě SW analýzy také volba vhodných technologií pro vývoj systému. Firma využívá rozmanité programovací jazyky, ale vzhledem k zachování technologií, které jsou běžně použity pro většinu webových systémů společnosti, padla volba na PHP5 + Zend Framework a MySQL databázi. K této serverové kombinaci programovacího jazyka, frameworku a databáze potom přibýly standardní webové technologie jako jsou CSS (pro formátování a vzhled), JavaScript pro asynchronní zpracování požadavků (AJAX) a dynamické úpravy v DOM modelu webové aplikace i pro další programování v prohlížeči. Jelikož se při programování používá standardně anglický jazyk, při návrhu systému i implementaci jsem se držel anglických pojmenování.

#### 3.1 Analýza

Analýza připravovaného nabídkového systému vychází z popisu prodejního procesu a zejména procesu vytváření nabídky z kapitoly 2. Nabídkový systém má poskytnout zákazníkovi co nejjednodušší a nejrychlejší formou potřebné informace k vybraným zájezdům a evidovat informace o nabídkách v prodejním systému. Z tohoto důvodu bude nabídkový systém spolupracovat s oběma těmito systémy. Ke spolupráci je vhodné využít webové služby, které tyto okolní systémy již poskytují.

Prodejce po přihlášení pod svým účtem na webu může vybírat vhodné zájezdy z celé databáze, kterou má k dispozici. Vybrané zájezdy označí pro použití v nabídce. V nabídkovém systému si prodejce zvolí vytvoření nové nabídky. Ten následně vyzve prodejce k volbě, ke které objednavce vytvářená nabídka patří. Seznam objednávek, které má prodejce k dispozici systém získá přes WS (webové služby) z prodejního systému. Po výběru objednávky si uloží k nabídce data o zákazníkovi i data o prodejci do vlastní databáze. Pomocí WS se zeptá webového portálu na konkrétní vybrané zájezdy a díky vhodným metodám získá data o těchto vybraných zájezdech. Tato data také uloží do databáze nabídkového systému. Tím je zajištěno, že data nabídky platná k danému časovému okamžiku nebudou ovlivněna změnou dat v databázi zájezdů a prodejce může data nabídky aktualizovat aniž by ovlivnil původní data zobrazovaná na webu.

Následně prodejce doplní informace do nabídky, přidá cenové kalkulace a sestaví text průvodního e-mailu doplněný odkazem na online nabídku. Ten bude zaslán zákazníkovi. Zároveň nabídkový systém pomocí WS zašle prodejnímu systému informaci o odeslání nabídky k dané objednávce. Zákazník si ve svém prohlížeči projde nabízené zájezdy včetně všech dostupných informací k nim a pokud ho některá z nabídek zaujme, jedním kliknutím zašle nabídkovému systému informaci o svém výběru. Nabídkový systém tuto skutečnost opět pomocí WS pošle do prodejního systému a ten notifikuje prodejce. Vazby jednotlivých systémů, databází a uživatelů znázorňuje následující schéma - obdélníky znázorňují jednotlivé systémy, symbol válce je znakem pro databázi.

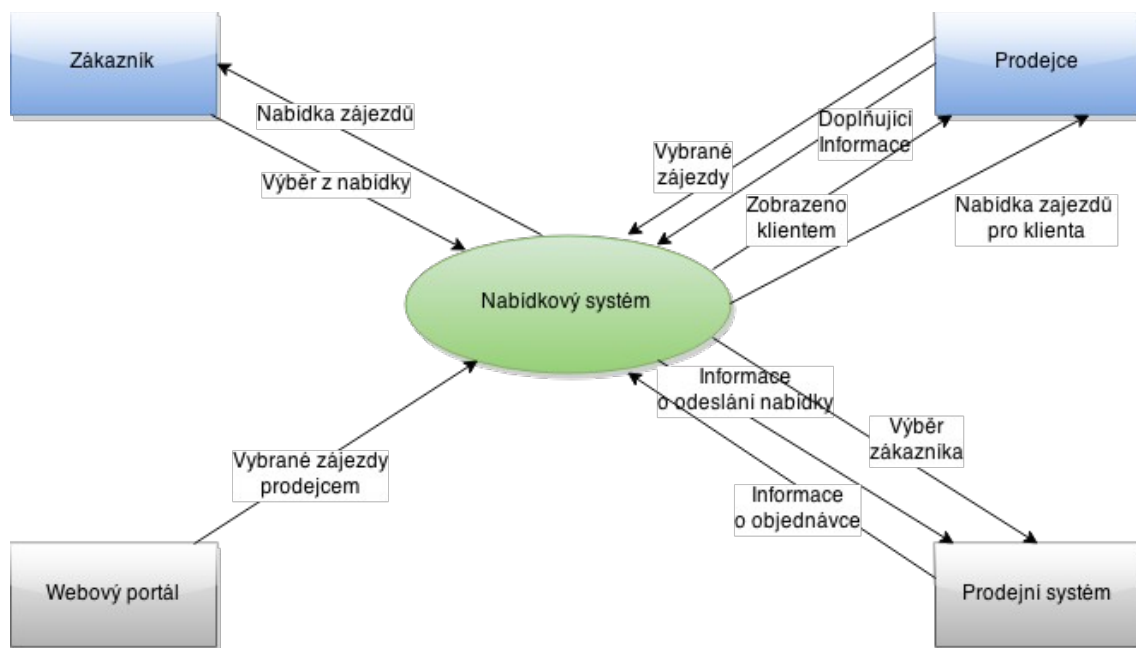


**Obrázek č. 5:** Schéma spolupráce nabídkového systému s ostatními systémy a databázemi. (Zdroj: Vlastní zpracování)

### 3.1.1 Kontextový diagram

Kontextový diagram je první úroveň tzv. Data Flow Diagramu (DFD). Celý systém je zde zobrazen jako jediný proces, který je obklopen tzv. terminátory, které představují okolí systému. Mezi procesem a terminátory jsou vyznačeny datové toky, které jsou orientovány směrem, kterým data putují a obsahují popisek o povaze dat. Kdybychom

chtěli rozvést kontextový diagram na podrobnější DFD diagram, rozdělili bychom hlavní proces na několik dílčích procesů a upravili datové toky k jednotlivým procesům. Pro účely této bakalářské práce si vystačím pouze z kontextovým diagramem na této úrovni.



Obrázek č. 6: Kontextový diagram pro nabídkový systém. (Zdroj: Vlastní zpracování)

### 3.1.2 Databáze

Dalším krokem analýzy je návrh datové struktury. Pro ukládání dat nabídkovému systému bude sloužit 6 relačních databázových tabulek. Atributy a vazby mezi tabulkami znázorňuje ERD diagram na obrázku č. 7. Modře vybarvené tabulky patří nabídkovému systému, zeleně vybarvené tabulky jsou uvedeny jen pro uvedení vazeb na tabulky okolních systémů, na které se nabídka váže.

Datový model obsahuje následující tabulky:

- *offer* - základní tabulka s vazbou na objednávku, prodejce i zákazníka, s textem e-mailu a časem posledního zobrazení nabídky
- *offer\_hotel* - hotely (zájezdy), které jsou obsaženy v nabídce

- *offer\_hotel\_photo* - fotografie hotelů, které se zobrazují v nabídce
- *offer\_hotel\_calculation* - kalkulace k hotelu
- *offer\_hotel\_calculation\_item* - cenová položka kalkulace
- *offer\_state* - číselník stavů nabídek



## Datový slovník:

Tabulka č. 3: Datový slovník tabulky *offer* (Zdroj: vlastní zpracování)

Sloupec	Typ	Komentář
offer_id	int(10)	číslo nabídky, primární klíč
user_id	int(10)	číslo prodejce
order_id	int(10)	číslo objednávky k nabídce
customer_id	int(10)	číslo zákazníka
state_id	int(10)	stav nabídky
created	timestamp	datum a čas vytvoření nabídky
valid_to	timestamp NULL	platnost nabídky do
mail_text	text NULL	průvodní text k nabídce
visited	timestamp NULL	datum a čas posledního zobrazení
visit_counter	int(11)	počet návštěv / zobrazení
visit_ip	varchar(255)	IP adresa posledního návštěvníka

Tabulka č. 4: Datový slovník tabulky *offer\_hotel* (Zdroj: vlastní zpracování)

Sloupec	Typ	Komentář
offer_hotel_id	int(10) unsigned	primární klíč
offer_id	int(11)	číslo nabídky, ke které hotel patří
hotel_id	int(11) NULL	číslo hotelu
name	varchar(255)	název hotelu
hotel_stars	double	počet hvězdiček
locality	varchar(255)	lokalita hotelu / zájezdu
start	timestamp	datum začátku zájezdu
end	timestamp	datum konce zájezdu
length	varchar(255)	délka zájezdu - počet dní
note	text NULL	poznámka k hotelu
transportation	varchar(255)	způsob dopravy
flight_departure	int(11) NULL	odletové místo
departure_place	varchar(255) NULL	odjezdové místo
meal	varchar(255) NULL	strava
accomodation	varchar(255) NULL	typ ubytování
google_maps_x	double NULL	x souřadnice GPS
google_maps_y	double NULL	y souřadnice GPS
tour_info	text NULL	text s informacemi o zájezdu
hotel_description	text NULL	popis hotelu
best_rating	tinyint(1) NULL	nejlepší hodnocení hotelu
discount_percent	tinyint(2) NULL	sleva v procentech
sortorder	tinyint(2) [99]	pořadí hotelu v nabídce

**Tabulka č. 5: Datový slovník tabulky *offer\_hotel\_calculation*** (Zdroj: vlastní zpracování)

Sloupec	Typ	Komentář
calc_id	int(11)	číslo kalkulace, primární klíč
offer_hotel_id	int(10) unsigned	číslo hotelu, ke kterému kalkulace patří
sequence	int(11)	pořadí kalkulace v rámci hotelu
name	varchar(255)	název kalkulace
note	text	poznámka ke kalkulaci
optional_currency_id	tinyint(4) NULL	číslo doplňkové měny

**Tabulka č. 6: Datový slovník tabulky *offer\_hotel\_calculation\_item*** (Zdroj: vlastní zpracování)

Sloupec	Typ	Komentář
calc_item_id	int(10) unsigned	číslo položky kalkulace, primární klíč
calc_id	int(11) unsigned	číslo kalkulace, ke které položka patří
name	varchar(255)	název položky kalkulace
description	varchar(255)	vysvětlující popis položky kalkulace
count	int(11)	zvolený počet této položky
price	double	cena za jednotku dané položky
currency_id	tinyint(4)	měna, která se váže k ceně

**Tabulka č. 7: Datový slovník tabulky *offer\_hotel\_photo*** (Zdroj: vlastní zpracování)

Sloupec	Typ	Komentář
offer_hotel_photo_id	int(10) unsigned	číslo fotografie hotelu, primární klíč
offer_hotel_id	int(10) unsigned	číslo hotelu, ke kterému fotografie patří
src	varchar(255)	cesta k fotografii
order	tinyint(4)	pořadí v nabídce
status	enum('visible','hidden')	viditelnost fotografie v nabídce
uploaded	boolean	uploadováno prodejcem nebo převzato z webu

**Tabulka č. 8: Datový slovník tabulky *offer\_state*** (Zdroj: vlastní zpracování)

Sloupec	Typ	Komentář
state_id	int(11)	číslo stavu, primární klíč
caption	varchar(255)	titulek názvu stavu

### 3.1.3 Návrh tříd

Diagram tříd uvedený na další straně ukazuje zjednodušenou strukturu objektových tříd a jejich vazby. Vzhledem k rozsáhlosti tříd a počtu metod není možné v rámci této práce uvést jejich kompletní přehled, proto diagram tříd obsahuje pro ukázkou jen některé z nich.

Třídy jsou v rámci návrhového vzoru model-view-controller rozděleny na modely a kontroler (controller). O část view z tohoto vzoru se stará komponenta *Zend\_View*, která je součástí Zend Frameworku, proto tato třída v mém návrhu tříd není uvedena.

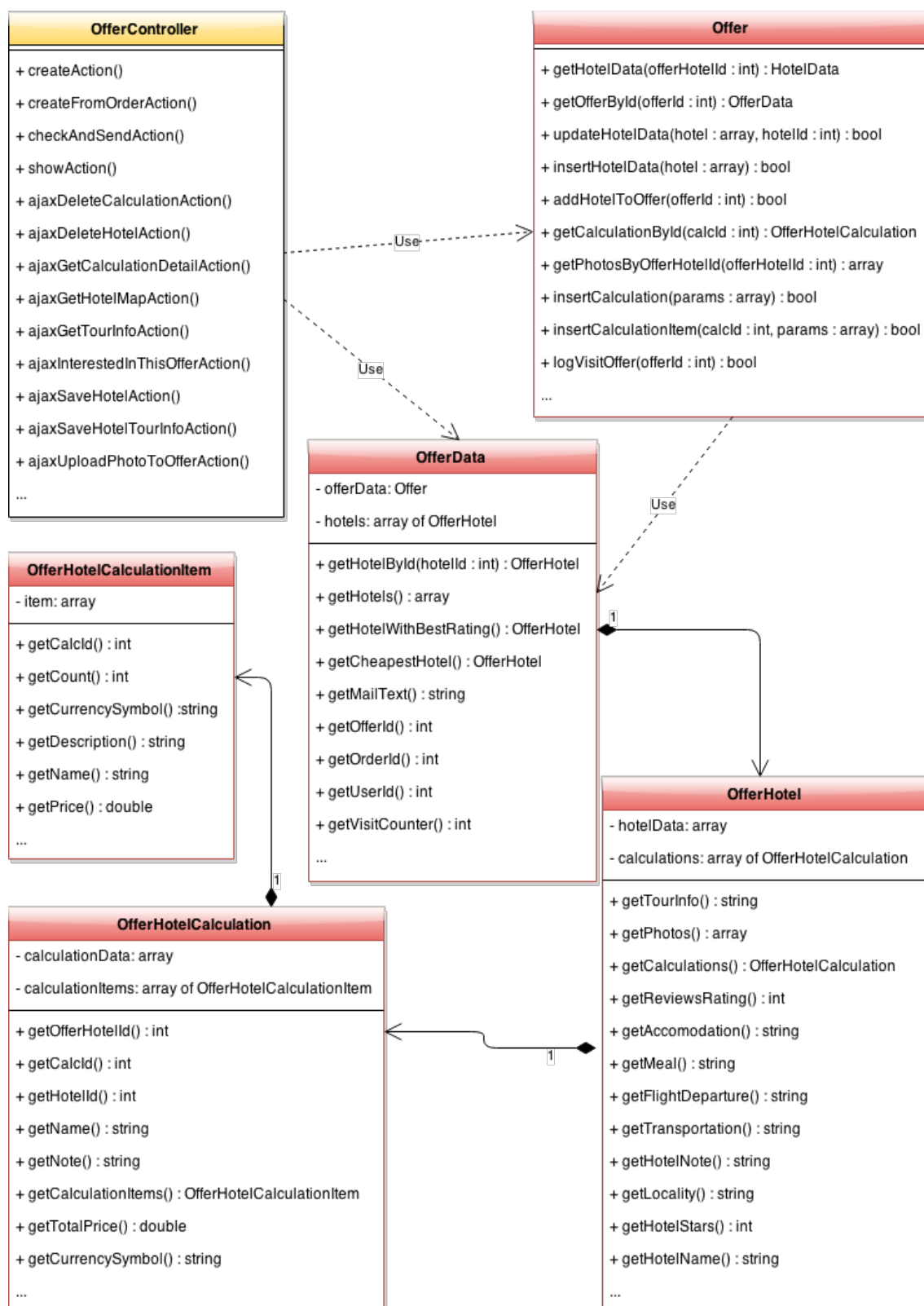
Kontroler obsahuje tzv. akce, což jsou metody z nichž každá řídí zobrazení dané stránky na určité adrese (URL). Název každé akce musí v Zend Frameworku končit suffixem "Action". Protože mnoho akcí je v nabídkovém systému zpracováváno asynchronně formou AJAXu, pro přehlednost názvy těchto akcí začínají prefixem "ajax".

V diagramu tříd můžeme vidět žlutě obarvenou třídu *OfferController*, která obsahuje akce *createAction()* pro vytvoření a editaci nabídky nebo *showAction()* pro zobrazení nabídky zákazníkovi. Ajaxové akce potom slouží pro vytváření kalkulací, editaci údajů k hotelu, zobrazení recenzí, mapy s hotelem, nahrávání doplňkových fotografií k nabídce nebo volbě vybrané nabídky zákazníkem.

Červeně obarvené třídy v diagramu znázorňují třídy patřící do vrstvy modelů. Model s názvem *Offer* slouží pro provádění databázových operací nad nabídkou. Zajišťuje tedy získávání, ukládání, editaci a mazání dat k nabídkám. Další znázorněné třídy jsou modely použité pro pohodlnou práci s daty a tvoří hierarchickou strukturu. Mezi těmito třídami existuje vztah kompozice - třída *OfferData* (základní data nabídky) obsahuje *OfferHotel* (informace o hotelu / zájezdu), ta obsahuje *OfferHotelCalculation* (informace o vytvořených kalkulacích) a každá *OfferHotelCalculation* obsahuje *OfferHotelCalculationItem* (samotnou položku kalkulaci).

Jak kontroler, tak model *Offer* potom využívá datový model *OfferData* pro práci s rozsáhlými daty, které každá nabídka obsahuje.





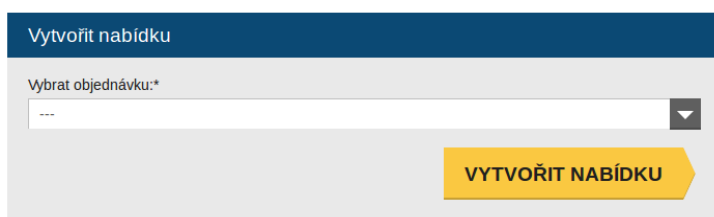
Obrázek č. 8: Diagram tříd s vybranými metodami. (Zdroj: vlastní zpracování)

## 3.2 Implementace

V této části popíšu samotnou implementaci a její části. Uvedu pro jednotlivé kroky názorné obrázky a kousky zdrojového kódu kontroleru. Vzhledem k rozsáhlosti systému a provázanosti s ostatními systémy není možné obsáhnout popis celého zdrojového kódu. Jen PHP třídy modelů a kontrolerů obsahují v součtu přes 5 tisíc řádků a nezahrnují šablony (view), JavaScript či formuláře. Systém je dostupný prodejci po přihlášení, pro něž se používá jednotná webová služba. Proto login není součástí této implementace.

### 3.2.1 Vytvoření nabídky

Pro vytvoření nové nabídky je potřeba vybrat objednávku, se kterou bude spojena. Po výběru objednávky systém na pozadí získá data o vybraných zájezdech k nabídce z webového portálu. Pro dotaz na tato data využije WS.



**Obrázek č. 9: Formulář pro založení nové nabídky.** (Zdroj: Vlastní zpracování)

Metoda *createFromOrderAction()* je akcí kontroleru, která se stará o zobrazení stránky obsahující formulář pro vytvoření nabídky. Nejprve zkontroluje, který z prodejců je právě přihlášený. Pokud žádný, přesměruje nabídkový systém stránku na přihlašovací obrazovku (volání privátní pomocné metody *getLoggedEmployeeOrRedirect()*). Následně je nastaven layout pro zobrazení šablony a vytvořena instance formuláře s výběrem objednávky prodejce. Pomocí WS je získán seznam aktivních objednávek prodejce, který je předán formuláři jako parametr metody *fillForm()*, která tyto položky do select boxu nastaví. Zkontrolujeme, zda nebyl již formulář odeslán a pokud ano, provedeme založení nové nabídky k vybrané objednávce zavoláním privátní pomocné metody *insertNewOffer()*. Na konci nastavíme instanci formuláře do proměnných pro šablonu, aby bylo možné jej vykreslit ve view - tzv. renderovat.

```

/**
 * Akce pro vyber z aktivnich objednavek uzivatele a vytvoreni nabidky
 * pri vytvoreni nabidky z webu
 */
public function createFromOrderAction()
{
    $currentUser = $this->getLoggedEmployeeOrRedirect();
    $this->_helper->layout->setLayout('layout-subpages-2col-left-wider');
    $user_id = $currentUser->getUserId();
    $form = new Form_Tour_Offer_ChooseOrder();
    $ws = $this->getServiceEmployee();
    $activeOrders = $ws->getUserActiveOrders($user_id);
    $tmpArray = array();
    foreach ($activeOrders as $ord) {
        $tmpArray[] = $ord['obj_id'];
    }

    array_multisort($tmpArray, SORT_DESC, $activeOrders);
    $options = array();
    $options[""] = '---';
    foreach ($activeOrders as $order) {
        $options[$order['nl_obj_id']] = $order['obj_id'] . ": " . $order['first']
            . " " . $order['lastname']
            . " (" . $order['email'] . ') - ' . $order['hotel'];
    }
    $form->fillForm($options);
    if ($this->getRequest()->isPost() && $form->isValid($this->getRequest()->getParams()))
    {
        $this->insertNewOffer($user_id, $this->getRequest()->getParam('order_id') );
    }
    $this->view->form = $form;
}

```

### 3.2.2 Editace nabídky

Po kliknutí na tlačítko “Vytvořit nabídku” a načtení zájezdů vybraných pro nabídku je zobrazena interaktivní obrazovka pro editaci. Prodejce vidí předvyplněná data z databáze a podle aktuálních upřesňujících informací je může upravit. Pod základními údaji o zájezdu jsou viditelné fotografie z daného místa a pod nimi je umístěno tlačítko pro editaci fotografií. Po kliknutí na toto tlačítko se zobrazí dialogové okno s možností aktivovat a deaktivovat fotografie k nabídce, popř. nahrát fotografii vlastní. Vedle fotografií je prostor pro přidání variant cenových kalkulací. Po kliknutí na “Přidat kalkulaci” je otevřeno vyskakovací okno, do kterého se přednačtou dostupné ceníkové

položky k danému zájezdu. Prodejce je opět může upravit, zadat počty osob, dětí apod. Ceny jsou automaticky přepočítávány podle zadanych hodnot v prohlížeči pomocí JavaScriptu.

**Nabídka zájezdu č. 1** 😊 7,3 Velmi dobré 46 hodnocení Pořadí: 1

Casa Nicola 2 Číslo zájezdu / Číslo termínu: 701983 / 52643216

Ischia Itálie

**Informace**

- Informace k zájezdu
- Recenze hotelu
- Oblast a mapa

**Poznámka od prodejce**

Tento zájezd můžu doporučit z vlastní zkušenosti.

**ULOŽIT ZMĚNY**

**Odstranit z nabídky**

**Cena za Váš zájezd celkem:**

<b>Varianta A:</b> 13 990 Kč	ABC	Editovat
<b>Varianta B:</b> 27 980 Kč	xxx	Editovat
<b>Varianta C:</b> 27 580 Kč	nova	Editovat
<b>Varianta D:</b> 27 980 Kč	nová 3	Editovat
<b>Varianta E:</b> 499 Kč		Editovat

**Přidat kalkulaci**

Termín zájezdu: 28.02.2014 - 07.03.14

Délka pobytu: 8 dní

Typ: Katalog

**Upravit fotografie**

☐ Naš tip

Sleva:

**Obrázek č. 10: Editace nabídky ze strany prodejce.** (Zdroj: Vlastní zpracování)

Následující zdrojový kód ukazuje metodu kontroleru s názvem *checkCalculations()*, která se zavolá vždy při přechodu do dalšího kroku vytvoření nabídkového e-mailu. Jelikož je volána pouze uvnitř kontroleru, jedná se o privátní metodu této třídy. Přijímá jako parametr *\$offer\_id*, což je číslo nabídky. Nejdříve získá data nabídky pomocí metody *getOfferById()* z třídy *Offer*. Získaný objekt *OfferData* obsahuje data celé nabídky. Následně prochází všechny hotely v této nabídce a na každém hotelu (instance třídy *OfferHotel*) volá metodu *hasAnyCalculation()*, která vrátí typ *boolean* - zda tento hotel v nabídce má vytvořenou alespoň jednu kalkulaci. Uvedená metoda se používá pro kontrolu, že všechny hotely v nabídce kalkulaci obsahují a pokud vrátí *false*, tak uživatel není do dalšího kroku procesu puštěn.

```

/**
 * Zkontroluje, zda jsou u všech hotelů kalkulace
 *
 * @param int $offer_id id nabídky
 * @return boolean
 */
private function checkCalculations($offer_id)
{
    $offer = $this->getModelTourOffer()->getOfferById($offer_id);

    $hotels = $offer->getHotels();
    foreach ($hotels as $hotel) {
        /* @var $hotel Model_Tour_Offer_OfferHotel */
        if (!$hotel->hasAnyCalculation()) {
            return false;
        }
    }
    return true;
}

```

### 3.2.3 Odeslání nabídky

#### Nabídkový e-mail

Odesílatel:\*

Příjemce:\*

Kopie (BCC):

Předmět:\*

Text:\*

Vážená paní, Vážený pane,

[Vaši nabídku naleznete zde.](#)

V rámci našich dalších služeb Vám doporučuji příkoupení komplexního cestovního pojištění. Cena pojištění pro Vámi vybraný pobyt je uvedena v kalkulaci zájezdu .

Uvedené ceny a podmínky nabídky jsou platné dle dnešní aktuální situace a mohou se v případě pozdější rezervace změnit.

V případě jakýchkoliv dotazů mne neváhejte kontaktovat na telefonním čísle .

S pozdravem a přáním krásného dne

**Obrázek č. 11: Sestavení textu e-mailu a odeslání zákazníkovi.** (Zdroj: Vlastní zpracování)

Dalším krokem a další částí nabídkového systému je sestavení průvodního e-mailu k nabídce s obsaženým odkazem na tuto nabídku na webu. Prodejci se zobrazí formulář s předvyplněným kontaktem na zákazníka (podle objednávky) a základním textem,

který může upravit či doplnit. K e-mailu je automaticky přidána patička s podpisem prodejce a následně nabídkový systém odešle nabídku zákazníkovi.

Níže uvedený útržek zdrojového kódu ukazuje přípravu e-mailu před jeho odesláním. Nejprve se pokusí získat e-maily oddělené čárkou nebo středníkem z textu zadaného do formulářového pole pro příjemce (proměnná *\$params['recipient']*). Pole může totiž obsahovat jak jeden e-mail, tak více e-mailů oddělených čárkou či středníkem. Následně je vytvořena instance třídy *Mail\_Tour\_OfferMail*, která je potomkem třídy pro práci s e-maily dostupné ve firemní knihovně, pouze přetěžuje některé její funkce pro účely nabídkového systému. Nyní projdeme pole získaných e-mailových adres a přidáme je mezi adresáty pomocí metody *addTo()*, kterou poskytuje třída *Mail\_Tour\_OfferMail* resp. *Zend\_Mail*. E-mailu nastavíme předmět, odesílatele, tělo zprávy a můžeme jej odeslat.

```
$params = $this->getRequest()->getParams();
$emails = $this->explodeEmails($params['recipient']);

$mail = new Mail_Tour_OfferMail();
foreach ($emails as $email) {
    $email = trim($email);
    if ($email) {
        $mail->addTo($email);
    }
}

$mail->setFrom($params['sender']);
$mail->setSubject($params['subject']);
$mail->setMailBody($params['mail_text']);
$mail->send();
```

### 3.2.4 Zobrazení nabídky zákazníkem

Zákazníkovi na e-mail přijde od prodejce zaslaný text s odkazem na detailní nabídku, která se zákazníkovi zobrazí ve webovém prohlížeči. Na této adrese je vidět kromě samotné nabídky (v podobném formátu jako vidí prodejce) také samotný průvodní e-mail a kontakt na prodejce, který nabídku zpracovával. Zákazník též automaticky vidí, který zájezd je nejlevnější, který je nejlépe hodnocený a který je nabízen s nejvyšší slevou. Také je zde viditelné tlačítko “Objednat” pro zvolení vybrané nabídky.

**Potřebujete poradit?**  
Vážená paní, Vážený pane,  
Vaši nabídku naleznete zde.

V nabídce zájezdu naleznete i nabídku cestovního pojištění, které vás ochrání v neočekávaných situacích po celou dobu Vaší dovolené.

Níže uvedené ceny a podmínky nabídky jsou platné dle dnešní aktuální situace a mohou se v případě pozdější rezervace změnit.

V případě jakýchkoliv dotazů mne neváhejte kontaktovat na telefonním čísle nebo na e-mail.

**V nabídce máte 6 zájezdů z toho**

- Nejlevnější zájezd: Casa Nicola
- Nejlépe hodnocený hotel: Villa Maria Clever
- Zájezd s nejvyšší slevou: Casa Nicola

Nabídka platná ke dni

**Nabídka zájezdu č. 1** 7,3 **Velmi dobré**  
62 hodnocení

**OBJEDNÁNÍ**

**Casa Nicola** ★★  
Ischia Itálie

**SLEVA 40 %**

**Další fotografie v galerii**

**Cena za Váš zájezd celkem:**

<b>Varianta A:</b> ABC	<b>13 990 Kč</b> Zobrazit
<b>Varianta B:</b> xxx	<b>27 980 Kč</b> Zobrazit
<b>Varianta C:</b> nova	<b>27 580 Kč</b> Zobrazit
<b>Varianta D:</b> nová 3	<b>27 980 Kč</b> Zobrazit
<b>Varianta E:</b>	<b>499 Kč</b> Zobrazit

**Termín zájezdu:** 28.02.2014 - 07.03.2014

**Délka pobytu:** 8 dní dní

**Typ ubytování:** Hotel

**Doprava:** Letecky

**Informace**

- Informace k zájezdu
- Recenze hotelu
- Oblast a mapa

**Poznámka od prodejce**

Tento zájezd můžu doporučit z vlastní zkušenosti.

Obrázek č. 12: Pohled na nabídku ze strany zákazníka. (Zdroj: Vlastní zpracování)

Útržek zdrojového kódu kontroleru zobrazuje část programu, která provádí získávání dat o prodejci a předání těchto údajů do view pro zobrazení v HTML šabloně. Nejprve získáme pomocí modelu *Offer* data ke konkrétní nabídce, následně vytvoříme instanci třídy, která realizuje komunikaci s WS pro uživatele a pomocí ní získáme data o prodejci, který nabídku vytvořil. Pokud uživatel nemá definovanou vlastní fotografii,

je mu nastaven výchozí avatar muže nebo ženy podle jeho pohlaví. Pak jsou všechna potřebná data nastavena jako proměnné pro view.

```
$offerModel = new Offer();
$offer = $offerModel->getOfferById($this->getRequest()->getParam('offer_id'));
$userService = $this->getServiceUser();

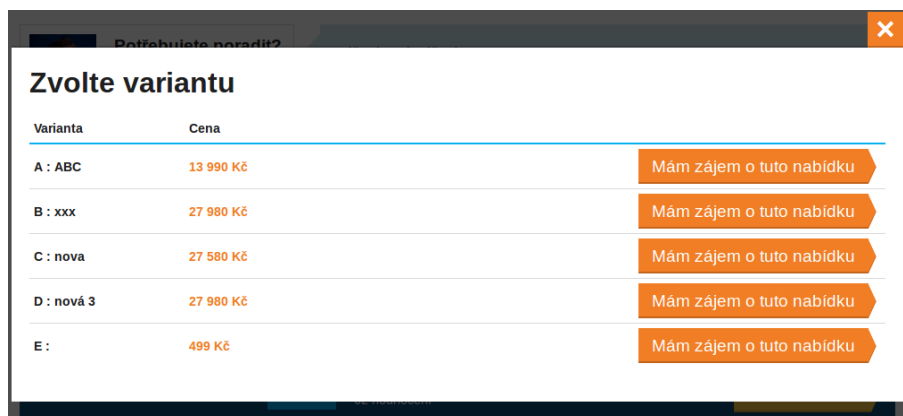
$wsUser = $userService->getSellerData($offer->getUserId());

$staticPhoto = null;
if (!$wsUser['photo']) {
    if ($wsUser['sex'] == Model_Tour_Offer::SEX_FEMALE) {
        $staticPhoto = $this->view->getHelper('Url')->getStatic('avatar_woman.jpg');
    } else if ($wsUser['sex'] == Model_Tour_Offer::SEX_MALE) {
        $staticPhoto = $this->view->getHelper('Url')->getStatic('avatar_man.jpg');
    }
} else {
    $staticPhoto = $wsUser['photo'];
}

$this->view->user = $wsUser;
$this->view->offer = $offer;
$this->view->staticPhoto = $staticPhoto;
$this->view->offer_id = $offer->getOfferId();
```

### 3.2.5 Reakce na nabídku

Jakmile zákazník klikne ve své nabídce na velké tlačítko “Objednat”, zobrazí se vyskakovací okno s přehledem všech cenových variant. U každé varianty má zákazník možnost vybrat, že “Má zájem o tuto nabídku”. V případě výběru potom nabídkový systém automaticky notifikuje prodejce o zákaznickové volbě.



Obrázek č. 13: Volba vybrané varianty z nabídky. (Zdroj: Vlastní zpracování)



Metoda *ajaxInterestedInThisOfferAction()* uvedena na další stránce je akcí kontroleru a je v tomto případě volána asynchronně pomocí AJAXu. Nejprve získá instanci třídy WS realizující komunikaci s prodejním systémem, následně dostaneme díky modelu *Offer* kalkulaci, jejíž unikátní číslo je předáno jako parametr této akci. Postupně získáme až číslo samotné nabídky, ke které tato kalkulace patří. Na závěr zašleme přes WS metodou *setChosenCalculation()* prodejnímu systému informaci, že k dané nabídce konkrétní objednávky byla zvolena určitá varianta kalkulace.

```
public function ajaxInterestedInThisOfferAction()
{
    $wsOffer = $this->getServiceOffer();
    $calc_id = $this->getRequest()->getParam('calc_id');
    $calc = $this->getModelTourOffer()->getCalculationById($calc_id);
    $offer_hotel_id = $calc->getOfferHotelId();
    $offerHotel = $this->getModelTourOffer()->getHotelDataArray($offer_hotel_id);
    $offer_id = $offerHotel['offer_id'];
    $offer = $this->getModelTourOffer()->getOfferById($offer_id);
    $order_id = $offer->getOrderId();

    $user_id = $offer->getUserId();
    $return = $wsOffer->setChosenCalculation($calc_id, $offer_id,
                                            $order_id, $user_id);

    echo $return;
    exit;
}
```

## 4 Ekonomické zhodnocení

V této kapitole se budu věnovat ekonomickému zhodnocení navrhnutého nabídkového systému. Budu zde zejména porovnávat předpokládané vynaložené finanční náklady spojené s vytvořením a očekávané přínosy. Jelikož se jedná o interní systém, ze kterého není viditelný přímý finanční výnos v tržbách, bude třeba toto hodnocení převést na odhadovaný přínos, který pro společnost bude jeho nasazení mít.

### 4.1 Náklady

Do nákladů vynaložených na nabídkový systém je třeba započítat práci, která je vynaložena na návrh a zpracování nabídkového systému. Je potřeba počítat jak práci analytika a programátora, tak práci ostatních členů týmu, kteří se účastní konzultací a testování. Budu předpokládat, že jsou to zaměstnanci firmy.

Práci budu měřit v tzv. člověkohodinách a ty ocením průměrnou hodinovou sazbou, kterou orientačně určím na základě dat dostupných v internetových celorepublikových průzkumech platů.<sup>32</sup> Z uváděné průměrné hrubé mzdy spočítám mzdové náklady firmy na zaměstnance (superhrubá mzda) a podle průměrného počtu pracovních hodin v měsíci určím průměrný hodinový náklad.

Zaměstnancovo vybavení (PC, energie) do nákladů nepočítám, neboť tvoří pro firmu fixní náklady na zaměstnance a neváže se na tento konkrétní projekt. Firma disponuje vlastní serverovou infrastrukturou, takže není možné jednoduše určit, jakou část z tohoto výkonu spotřebuje na provoz nabídkového systému. Proto do uvažovaných nákladů zahrnu průměrnou cenu virtuálního serveru od společností, která jej na našem trhu nabízí.

Předpokládané náklady shrnují tabulky na další straně.

---

<sup>32</sup> <http://www.platy.cz/>

**Tabulka č.9: Předpokládané náklady na vývoj nabídkového systému a první rok provozu.** (Zdroj: Vlastní zpracování)

Práce / služba	Množství	Částka za jedn. (Kč)	Částka celkem (Kč)
Analýza a vývoj	51 člh	250,-	127 500,-
Testování	51 člh	150,-	7 650,-
Virtuální server / měsíc	12 měsíců	1500,-	18 000,-
Doména / rok	1 rok	300,-	300,-
<b>Celkem:</b>			<b>153 150,-</b>

**Tabulka č.10: Předpokládané náklady na druhý a další rok provozu.** (Zdroj: Vlastní zpracování)

Práce / služba	Množství	Částka za jedn. (Kč)	Částka celkem (Kč)
Virtuální server / měsíc	12 měsíců	1500,-	18 000,-
Doména / rok	1 rok	300,-	300,-
<b>Celkem</b>			<b>18 300,-</b>

Za první rok, ve kterém je nabídkový systém vytvořen se tedy předpokládané náklady vyšplhají na 153 150,- Kč a následující roky bude stát provoz 18 300,- Kč v případě, že nevzniknou žádné požadavky na další programátorské úpravy či rozšíření.

## 4.2 Přínosy

Na přínosy se můžeme podívat ze dvou hledisek: nabídkový systém zlepšuje vnímání firmy zákazníkem (posiluje značku) a šetří čas prodejcům. Zatímco vnímání firmy zákazníkem je jako přínos těžké finančně ohodnotit, ušetřený čas vyčíslit lze. Předpokládám, že nabídkový systém ušetří prodejci 10 minut práce při vytvoření jedné nabídky. Kdyby prodejce měsíčně vytvořil pouze 15 nabídek, potom by šlo o ušetřených 500 hodin měsíčně u všech prodejců. Např. při hodinové mzdě 150,- Kč<sup>33</sup> se potom v daném měsíci jedná o ušetřený časový náklad v hodnotě 75 000,- Kč za měsíc.

<sup>33</sup> <http://www.platy.cz/platy/cestovni-ruch-gastronomie-hotelnictvi/prodejce-letenek-zajezdu>

**Tabulka č.11: Ušetřené náklady nabídkovým systémem.** (Zdroj: Vlastní zpracování)

Práce / služba	Čas	Počet	Ušetřený čas celkem
Ušetřený čas nad nabídkou.	10 minut / prodejce	200	33,33 hod.

Dá se tedy předpokládat, že by se náklady na nabídkový systém měly vrátit hned v průběhu prvního roku po spuštění (podle počtu realizovaných nabídek) a následně by měl již šetřit náklady. Snížení časových nákladů na zpracování nabídky může vést buď ke snížení finančních nákladů nebo k možnosti uspokojit nabídkou více klientů, přičemž oba dva efekty jsou žádoucí.

## ZÁVĚR

Cílem této práce bylo navrhnout nabídkový systém, který by fungoval jako online aplikace ve webovém prohlížeči. Bylo potřeba sjednotit firemní výstupy pro všechny zasílané nabídky a automatizovat a zefektivnit její vytvoření. Jelikož firma stojí na několika informačních systémech, které obsahují data potřebná pro vytváření klientských nabídek, bylo nutné, aby tento nabídkový systém dokázal s těmito systémy komunikovat a získávat od nich informace.

Proto jsem se v teoretické části zaměřil na popis technologií, které jsou využívány při vývoji webových aplikací a podle požadavků na nabídkový systém byly použity i při vývoji tohoto systému. Nejdříve jsem objasnil základní principy objektově orientovaného programování a s ním související téma návrhových vzorů. Následně jsem představil skriptovací jazyk PHP, Zend Framework, dále potom databázový systém MySQL. Podrobněji jsem rozebral tzv. webové služby, které jsou v nabídkovém systému široce využity.

Následující kapitola se zabývá popisem firmy, její struktury, přehledem nabízených produktů a služeb. Analyzuje prodejní proces a proces vytváření nabídek, zmiňuje informační systémy, které firma využívá. V závěru byla vypracována SWOT analýza společnosti.

V části určené pro návrh řešení jsem nejdříve zvolil technologie, které budou pro řešení využity a následoval analýzou navrhovaného nabídkového systému. Nejdříve bylo nutné popsat prostředí, ve kterém systém bude fungovat a navrhnout vazby na okolní systémy. Následně jsem v kontextovém diagramu znázornil datové toky, které budou plynout do systému a z něj. Na základě charakteru určených datových toků bylo možno detailně navrhnout datovou strukturu pro nabídkový systém a popsat jej formou ERD diagramu a datového slovníku. Na základě analýzy datové struktury bylo možné navrhnout základní objektové třídy aplikace, především tzv. modely. Pro zmapování základní struktury těchto tříd a jejich vazby na kontroler aplikace jsem zvolil diagram tříd. Takto navržený systém bylo možné následně podle analýzy implementovat. Tento krok je v práci dokumentován názornými obrázky ze systému a komentovanými částmi zdrojového kódu. Mezi hlavní popsané části vytvořeného nabídkového systému patří

vytvoření nabídky, editace nabídky a jejích dat, odeslání nabídky zákazníkovi, zobrazení nabídky zákazníkem a jeho reakce na nabídku - objednávka konkrétní varianty.

Další kapitola se věnuje ekonomickému zhodnocení realizovaného projektu, kde jsem se snažil vyčíslit rámcově náklady na systém a očekávané přínosy jeho využívání.

Popsaný nabídkový systém se tedy podařilo s úspěchem navrhnout a implementovat do prostředí společnosti tak, že je aktivně využíván a plní požadavky, které na něj byly kladeny. Z části automatizuje proces vytváření nabídek a sjednocuje komunikaci směrem k zákazníkům. Hned při testovacím nasazení přes něj bylo zasíláno řádově stovky nabídek a po plném nasazení se tento počet ještě mnohonásobně navýšil.

## SEZNAM POUŽITÝCH ZDROJŮ

1. AL], Cristian Darie ... [et]. AJAX and PHP building responsive web applications. Birmingham, U.K: Packt Pub, 2006. ISBN 19-048-1182-5.
2. BUDD, Andy a With Cameron Moll and Simon COLLISON. CSS mastery advanced web standards solutions. New ed. Berkeley [Calif.]: Apress, 2006. ISBN 15-905-9614-5.
3. ERL, Thomas. *SOA: servisně orientovaná architektura : kompletní průvodce*. Vyd. 1. Překlad Ondřej Baše, Lukáš Krejčí. Brno: Computer Press, 2009, 671 s. ISBN 978-80-251-1886-3.
4. GUTMANS, Andi. Mistrovství v PHP 5. Vyd. 2. Brno: Computer Press, 2007, 655 s. ISBN 978-80-251-1519-0.
5. KEITH, Jeremy. HTML5 FOR WEB DESIGNERS. 1. vyd. New York: A Book Apart, 2010. ISBN 978-0-9844425-0-8.
6. KOFLER, Michael. Mistrovství v MySQL 5. Vyd. 1. Překlad Jan Svoboda, Ondřej Baše, Jaroslav Černý. Brno: Computer Press, 2007, 805 s. ISBN 978-80-251-1502-2.
7. PECINOVSKÝ, Rudolf. Návrhové vzory. Vyd. 1. Brno: Computer Press, 2007, 527 s. ISBN 978-80-251-1582-4.
8. WELLING, Luke. PHP a MySQL - rozvoj webových aplikací. Praha: SoftPress, 2002, 718 s. ISBN 80-864-9720-8.

### Internetové zdroje:

1. ADAPTIC, S.R.O., Affiliate marketing [online]. [cit. 2015-01-26]. Dostupné z: <http://www.adaptic.cz/znalosti/slovnicek/affiliate-marketing/>
2. Git --local-branching-on-the-cheap [online]. [cit. 2015-02-20]. Dostupné z: <https://git-scm.herokuapp.com/>
3. HORDĚJČUK, Vojtěch. Objektově orientované programování [online]. [cit. 2015-05-26]. Dostupné z: <http://voho.cz/wiki/oop/>
- 4.
5. LMC, S.R.O., Prodejce letenek/zájezdů [online]. [cit. 2015-05-20]. Dostupné z: <http://www.platy.cz/platy/cestovni-ruch-gastronomie-hotelnictvi/prodejce-letenek-zajezdu>

6. LMC, S.R.O.,. Programátor PHP [online]. [cit. 2015-05-20]. Dostupné z: <http://www.platy.cz/platy/informacni-technologie/programator-php>
7. ORACLE CORPORATION,. MySQL Community Edition [online]. [cit. 2015-02-27]. Dostupné z: <http://www.mysql.com/products/community/>
8. REFSNES DATA,. CSS Tutorial [online]. [cit. 2015-03-27]. Dostupné z: <http://www.w3schools.com/css/default.asp>
9. ŠTRAUCH, Adam. Databáze MariaDB válcuje MySQL [online]. [cit. 2015-03-14]. Dostupné z: <http://www.root.cz/clanky/databaze-mariadb-valcuje-mysql/>
10. REFSNES DATA,. PHP MySQL Database [online]. [cit. 2015-02-25]. Dostupné z: [http://www.w3schools.com/php/php\\_mysql\\_intro.asp](http://www.w3schools.com/php/php_mysql_intro.asp)
11. THE JQUERY FOUNDATION,. JQuery - write less, do more [online]. [cit. 2015-02-27]. Dostupné z: <https://jquery.com/>
12. VÁCLAVEK, Petr. AJAX a PHP – tvoříme interaktivní webové aplikace profesionálně [online]. [cit. 2015-03-10]. Dostupné z: <https://www.interval.cz/clanky/ajax-a-php-tvorime-interaktivni-webove-aplikace-profesionalne/>
13. W3C,. SOAP Version 1.2 Part 1: Messaging Framework [online]. [cit. 2015-01-31]. Dostupné z: <http://www.w3.org/TR/soap12-part1/>
14. ZEND TECHNOLOGIES LTD,. About [online]. [cit. 2015-02-21]. Dostupné z: <http://framework.zend.com/about/>
15. ZEND TECHNOLOGIES LTD,. Create Your Project [online]. [cit. 2015-02-21]. Dostupné z: <http://framework.zend.com/manual/1.12/en/learning.quickstart.create-project.html>
16. ZEND TECHNOLOGIES LTD,. Overview [online]. [cit. 2015-02-21]. Dostupné z: <http://framework.zend.com/manual/1.12/en/introduction.overview.html>
17. ZEND TECHNOLOGIES LTD,. Zend Controller Quick Start [online]. [cit. 2015-02-21]. Dostupné z: <http://framework.zend.com/manual/1.12/en/zend.controller.quickstart.html>
18. ZEND TECHNOLOGIES LTD,. Zend Framework Reference [online]. [cit. 2015-02-21]. Dostupné z: <http://framework.zend.com/manual/1.12/en/reference.html>



19. ZEND TECHNOLOGIES LTD,. Zend\_Cache [online]. [cit. 2015-02-21].  
Dostupné z: <http://framework.zend.com/manual/1.12/en/zend.cache.html>
20. ZEND TECHNOLOGIES LTD,. Zend\_Db\_Adapter [online]. [cit. 2015-02-21].  
Dostupné z: <http://framework.zend.com/manual/1.12/en/zend.db.adapter.html>

## SEZNAM OBRÁZKŮ

Obrázek č. 1: Základní struktura zprávy SOAP.....	23
Obrázek č. 2: Schéma hlavních systémů použitých ve společnosti.....	33
Obrázek č. 3: Schéma třívrstvé architektury webové aplikace.....	33
Obrázek č. 4: Vývojový diagram prodejního procesu.....	35
Obrázek č. 5: Schéma spolupráce nabídkového systému s ostatními systémy a databázemi.....	41
Obrázek č. 6: Kontextový diagram pro nabídkový systém.....	42
Obrázek 7: ERD diagram ukazující databázové tabulky a jejich vztahy.....	44
Obrázek č. 8: Diagram tříd s vybranými metodami.....	48
Obrázek č. 9: Formulář pro založení nové nabídky.....	49
Obrázek č. 10: Editace nabídky ze strany prodejce.....	51
Obrázek č. 11: Sestavení textu e-mailu a odeslání zákazníkovi.....	52
Obrázek č. 12: Pohled na nabídku ze strany zákazníka.....	54
Obrázek č. 13: Volba vybrané varianty z nabídky.....	55

## SEZNAM TABULEK

Tabulka č. 1: Základní údaje o firmě.....	31
Tabulka č. 2: SWOT analýza podniku.....	37
Tabulka č. 3: Datový slovník tabulky offer.....	45
Tabulka č. 4: Datový slovník tabulky offer_hotel.....	45
Tabulka č. 5: Datový slovník tabulky offer_hotel_calculation.....	46
Tabulka č. 6: Datový slovník tabulky offer_hotel_calculation_item.....	46
Tabulka č. 7: Datový slovník tabulky offer_hotel_photo.....	46
Tabulka č. 8: Datový slovník tabulky offer_state.....	46
Tabulka č. 9: Předp. náklady na vývoj nabídkového systému a první rok provozu.....	58
Tabulka č. 10: Předp. náklady na druhý a další rok provozu.....	58
Tabulka č. 11: Ušetřené náklady nabídkovým systémem.....	59

## SEZNAM POUŽITÝCH ZKRATEK A SYMBOLŮ

OOP	Object Oriented Programming (Objektově orientované programování)
HTML	HyperText Markup Language
XML	eXtensible Markup Language
SOAP	Simple Object Access Protocol
SW	software
MVC	model-view-controller
ZF	Zend Framework
MySQL	My Structured Query Language
GPL	General Public Licence
WSDL	Web Services Description Language
W3C	World Wide Web Consortium
DOM	Document Object Model
CSS	Cascading Style Sheets
AJAX	Asynchronous JavaScript and XML
CK	Cestovní kancelář
CA	Cestovní agentura
SWOT	Strengths, Weaknesses, Opportunities, Threats
ICT	Information and Communication Technology
WS	webové služby (WebServices)
DFD	Data Flow Diagram
ERD	Entity Relationship Diagram